Project No. 12-4065

An Integrated Fuel Depletion Calculator for Fuel Cycle Options Analysis

Mission Supporting Transformative Research

Erich Schneider
University of Texas, Austin

In collaboration with:
University of Wisconsin, Madison

Jack Wheeler, Federal POC
Temi Taiwo, Technical POC

NEUP Nuclear Energy University Programs
U.S. Department of Energy

NEUP 12-4065: An Integrated Fuel Depletion Calculator for Fuel Cycle Options Analysis.

Final Report

E. Schneider, R. Flanagan, C. Bagdatlioglu, UT-Austin and A. Scopatz, U. Wisconsin

Introduction

Bright-lite is a reactor modeling software developed at the University of Texas Austin to expand upon the work done with the Bright [1] reactor modeling software. Originally, Bright-lite was designed to function as a standalone reactor modeling software. However, this aim was refocused to couple Bright-lite with the Cyclus fuel cycle simulator [2] to make it a module for the fuel cycle simulator.

The remainder of the introduction summarizes our milestones and provides commentary on selected milestone goals. The methodology used in Bright-lite is presented in the next section. The specifics of different functionalities of the Bright-lite software can be seen in the Papers section of this work.

Milestones	Deliverable
Task 1: Implement baseline capability as a Cyclus module - **100% complete**	Infrastructure, validation, release
Task 2: Uncertainty capable library specification **- 100% complete**	
Task 3: Swappable interpolation method and recipe schema comparison **- 100% complete**	Comparison report of various schema
Task 4: Automated library validation technique **- 100% complete**	Report, release
Task 5: Optimization & Parallelism **- 100% complete**	(Reporting with Task 6)
Task 6: Serpent linkage, User interface, second release **- 100% complete**	Software release and documentation
Task 7: Multi-region macro geometry and embedded diffusion equation solver **- 100% complete**	Four reactor family cross section libraries
Task 8: Fully integrated Serpent linkage **- 100% complete**	
Task 9: Final release and documentation - **100% complete**	Software release and documentation

Milestone 1

The Bright-lite reactor capabilities have been incorporated into the Cyclus v1.2 frame work as a module. Given direction from DOE, this task became a large component of our work as we underwent several iterations with the Cyclus team at U Wisconsin. The iterations provided input to the Wisconsin group that helped them refactor the Cyclus core to become more amenable to coupling with externally-developed modules. Our module became the first external (to the Cyclus core team) module coupled to the simulator.

Milestone 2

This milestone was completed as a preliminary to Milestone 4. The Bright-lite software is distributed with several libraries. In addition, the library format is specified and users are encouraged to provide their own libraries generated for specific reactors in the studied fuel cycle. During reactor deployment the user can specify the library to be used for the reactor. In addition, the user can list libraries to be interpolated based on fuel cycle parameters (such as fuel composition, conversion ratio, or burnup).

Milestone 3

The completion of this milestone has two components. First, cross section library generation has been incorporated into the Bright project. Second, UT-Austin has completed a literature review of methodologies for reactor material recipe generation implemented in other simulation codes (e.g., VISION, DANESS, COSI, NFCSim).

Milestone 4 & 5 & 6

These three milestones are reported together because they are all part of a single workflow. The full integration of the library interpolation system into the core code of Bright-lite has been completed. The interpolation system is also used to generate new libraries to ensure that the interpolation method is always valid to a user specified tolerance. Libraries are validated during generation.

The user interface portion of milestone 6 is incorporated in milestone 1, as part of the linking with Cyclus.

Milestone 7
For this milestone a new reactor module for Cyclus was created called ReactorX. This distinction was made to handle the increased complexity of higher fidelity reactor modeling. ReactorX allows for region-level cross-section libraries to handle core designs incorporating multiple fuel types.

Milestone 8
The XSGen software which links ORIGEN with OpenMC (which was chosen in place of Serpent) has been completed in full. Using XSGen Bright-lite libraries can be generated automatically.

Milestone 9
The user guide to Bright-lite can be found in Appendix F of this document. This includes directions for installation and use.

Papers, Conference Proceedings, and Graduate Work

C. Bagdatlioglu, “Fuel Composition Generation Techniques of Nuclear Fuel Cycle Simulators,” Presented at 2014 American Nuclear Society Student Conference, State College, PA, April 4, 2014.

R. Flanagan, E. Schneider, C. Bagdatlioglu, "Fuel Composition Calculation Techniques of Nuclear Fuel Cycle Simulators," Physics of Reactors Conference (PHYSOR) 2014, The Westin Miyako, Kyoto, Japan, September 28 – October 3, 2014, on CD-ROM (2014)

R. Flanagan, E. Schneider, C. Bagdatlioglu, "Multidimensional Cross Section Library Interpolation for the Bright-lite Reactor Modeling Software," Global Fuel Cycle Conference, 2015.

C. Bagdatlioglu, E. Schneider, R. Flanagan, "Using Spatial Flux Calculations to Improve the Fluence-Based Neutron Balance Approach", Presented 2015 ANS Annual Meeting.

C. Bagdatlioglu, "Fluence Based Neutron Balance Approach Using Spatial Flux Calculations" August 2015, Master's Thesis, University of Texas Austin.

R. Flanagan, "Novel Methods for Generalizing Nuclear Fuel Cycle Design, and Fuel Burnup Modeling" December 2015, Dissertation, University of Texas Austin.

C. Bagdatlioglu, R. Flanagan, E. Schneider, "Characterizing the United States Nuclear Used Fuel Using Medium Fidelity Reactor Modeling Software," ICONE 24, Charlotte, NC, June 26-30 2016 (Accepted)

C. Bagdatlioglu, R. Flanagan, E. Schneider, "Fuel Cycle Analysis Using Bright-lite in the Cyclus Simulator," Physics of Rectors Conference (PHYSOR) 2016, Sun Valley, Idaho, May 1-5. (Accepted)

C. Bagdatlioglu, E. Schneider, R. Flanagan, "Method for Generating Reactor Material Balances for Fuel Cycle Simulations", Nuclear Engineering and Design (accepted with revisions), 2016.

Bright-lite methodology

Bright-lite is a Cyclus module developed to be a reactor isotopics and burnup calculator. It is designed to support two operating modes: a 'forward' mode in which it solves for output fuel isotopics and burnup given input isotopics, and a 'blending' mode where it finds the input fuel composition needed to achieve a target burnup given fuel input streams available for blending. The input fuel composition or, for the blending mode, the target burnup and fuel streams available for blending are supplied by Cyclus.

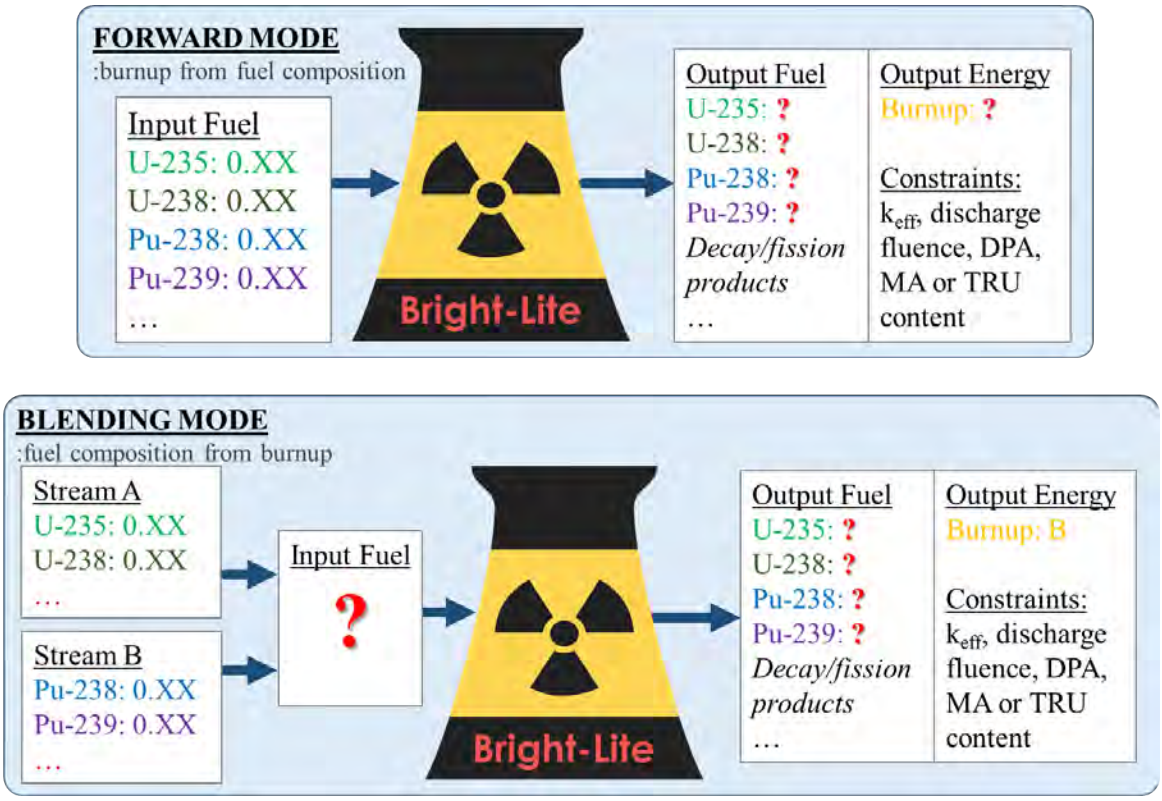

Figure 1 – Two modes of operation for Bright-lite.

Figure 1 depicts the two operating modes. The top panel of Figure 1 shows the forward mode where a known fuel composition is used to determine the unknown discharge fuel composition and burnup. The bottom panel illustrates the blending mode with two streams of known composition available for blending to fabricate fuel. In this case Bright-lite determines the input fuel composition (blending ratio of the two streams A and B) that would result in the given burnup.

Bright-lite is capable of imposing additional constraints aside from discharge burnup. For instance, a limit may be placed on the fluence to which the fuel can be exposed, or a target conversion ratio may be set. Since these other constraints are not a focus here, the methodology will be presented for the case where the discharge burnup is the only active constraint on the cycle length.

The methodology for finding the discharge burnup from a determined fuel composition is explained first. Next, the iterative method to determine input fuel composition to achieve a target burnup is detailed.

Isotope Library Creation

Bright-lite uses one energy group cross section libraries to pre-calculate and parameterize its burnup and transmutation calculations that are used to determine discharge burnup. The results of these

calculations are saved in isotope library databases for use during runtime. The database contains isotope libraries which are generated for every isotope that may be present in initial fuel loadings. Each reactor condition and fuel type has an associated one-group cross section set, and every one-group cross section set gives rise to its own isotope library database.

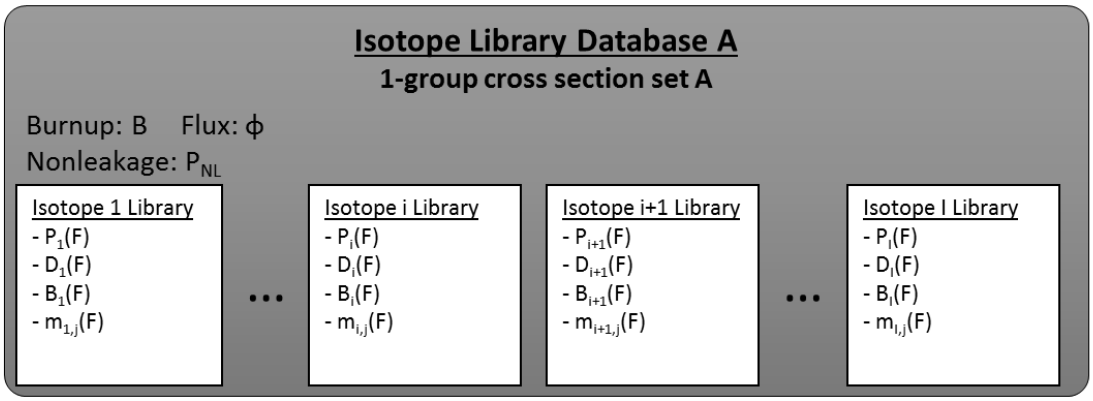

Figure 2 – Example isotope library database and isotope libraries schematic.

The isotope library database and its constituents are shown in Figure 2. Each isotope library stored in this database consists of the following vectors and matrix, all generated by exposing a unit mass of the isotope to a neutron flux (φ).

Each entry in the isotope library is recorded at distinct fluence values F_k, where k=1…K indexes the fluence up to the highest tabulated fluence level, F_K. This fluence is chosen to comfortably exceed the fluence attained at discharge by any realistic fuel. Even though values in the isotope library are recorded as discrete values corresponding to their discrete fluence, this report presents them as continuous functions. Linear interpolation is used to calculate intermediary values as described later in this section.

Isotopes in the isotope library database are represented with the index i=1…I, where 'I' is the total number of isotopes in the database. The transmutation and decay products are represented with the index j=1…J. 'J', the total number of transmutation and decay products, will be greater than or equal to 'I,' since 'J' includes fission products and other isotopes which may not be in initial fuel.

The isotope library for isotope 'i' contains the following:

- Fluence vector, F. The time integral of neutron flux. [n/cm2]
- Neutron production rate vector as a function of fluence, $P_i(F)$. [n/s/φ/kgIHM]
- Neutron destruction rate vector as a function of fluence, $D_i(F)$. [n/s/φ/kgIHM]
- Total burnup vector as a function of fluence, $B_i(F)$. [MWd/kgIHM]
- Isotope transmutation product matrix as a function of fluence, $m_{i,j}(F)$. The index j describes all the tracked isotopes that the initial isotope i transmutes or decays into. All decay and transmutation daughters of the initial mass are implicitly accounted for in the matrix, and their effects on the neutron economy are accounted in $P_i(F)$ and $D_i(F)$. [kg/kgIHM]

Bright-lite uses an existing depletion calculation tool (such as ORIGEN [3]) to determine the isotope libraries. An example isotope library for an U235 isotope in a LWR is presented in Table 1. Looking at this table, it can be observed that the isotope is transmuted to its activation and fission product daughters as flux is applied and fluence increased. As the parent U235 isotope is depleted, a decrease in neutron production rate is observed with increasing fluence. The neutron destruction rate per unit flux decreases as well because the U235 daughters generally have lower absorption cross sections and hence reaction rates than the parent.

Vector	**Units**	**F_1**	**F_2**	**F_3**	**F_4**	**F_5**	
F	[n/cm2]	7.78E+19	2.59E+21	5.70E+21	8.81E+21	1.19E+22	…
P(F)	[n/s/ϕ/kgIHM]	289	250	216	186	160	…
D(F)	[n/s/ϕ/kgIHM]	152	136	121	108	95.9	…
B(F)	[MWd/kgIHM]	3.5	108.5	201.7	282.1	351.4	…
Transmutation Matrix							
Isotope	**Units**	**F_1**	**F_2**	**F_3**	**F_4**	**F_5**	
U235	[g/kgIHM]	9.956E+02	8.623E+02	7.435E+02	6.411E+02	5.528E+02	…
U236	[g/kgIHM]	8.148E-01	2.505E+01	4.615E+01	6.385E+01	7.864E+01	…
U238	[g/kgIHM]	2.563E-07	2.525E-03	9.646E-03	2.061E-02	3.478E-02	…
PU238	[g/kgIHM]	2.464E-09	5.812E-03	4.666E-02	1.473E-01	3.230E-01	…
PU239	[g/kgIHM]	2.316E-12	1.318E-04	1.936E-03	8.516E-03	2.330E-02	…
PU240	[g/kgIHM]	3.094E-15	4.023E-06	1.141E-04	7.365E-04	2.630E-03	…
BA138	[g/kgIHM]	1.427E-01	4.496E+00	8.374E+00	1.172E+01	1.460E+01	…
CE142	[g/kgIHM]	1.244E-01	4.004E+00	7.463E+00	1.045E+01	1.302E+01	…
LA139	[g/kgIHM]	1.342E-01	4.290E+00	7.981E+00	1.115E+01	1.388E+01	…
MO97	[g/kgIHM]	5.967E-02	2.706E+00	5.063E+00	7.091E+00	8.837E+00	…
XE136	[g/kgIHM]	2.234E-01	7.480E+00	1.393E+01	1.949E+01	2.429E+01	…
ZR96	[g/kgIHM]	9.318E-02	2.895E+00	5.391E+00	7.543E+00	9.397E+00	…
⋮		⋮	⋮	⋮	⋮	⋮	⋯

Table 1 – U235 isotope library from a LWR Reactor database as a function of fluence.

To further illustrate the libraries, Figure 3 compares P(F) and D(F) for U-235 and U-238 created using a one-group cross section library for a generic PWR with 50 MWd/kg discharge burnup. The magnitude of the curve for U-235 (left) is significantly larger, and this isotope starts with a surplus of neutrons. The curve for U-238 increase in magnitude due to the breeding of new isotopes, notably Pu-239, which contribute to the neutron economy.

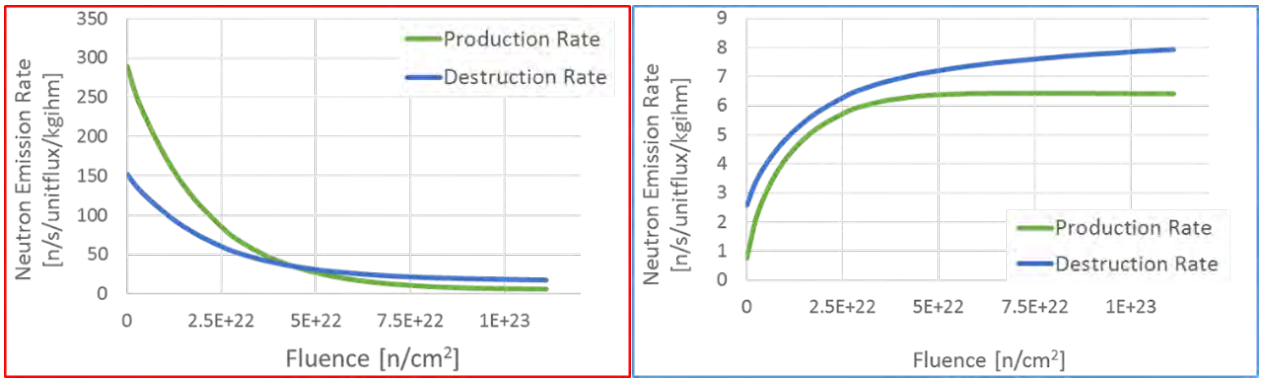

Figure 3 – Neutron production and destruction rates of U-235 (left) and U-238 (right).

This report assumes that *P(F)*, *D(F)*, *B(F)*, and $m_j(F)$ are defined as continuous functions. However, these functions are in practice obtained at discrete fluence values F_k, hence they are stored as discrete vectors. In the implementation of Bright-lite, intermediate values between discrete points are obtained using linear interpolation. The neutron production rate at fluence F (P(F)) can be used as an example to demonstrate this interpolation process. The subscript k refers to the discrete fluence step, such that

$$P^{-1}(P(F_k)) = F_k$$

To calculate an intermediate value at a fluence F falling between two tabulated fluences, the index k for F_k is found so that F is between F_k and F_{k+1}. Then the interpolation proceeds as follows:

$$P(F) = P(F_k)_f + (p(F_{k+1})_{f+1} - p(F_k))(F - F_k)/(F_{k+1} - F_k) \quad (1)$$

Fuel Library Creation from Known Composition

The isotope libraries in the database are used to generate a composition-specific fuel library, as shown in Figure 4. The one group cross section libraries and the isotope library databases derived from them can be specific to many parameters such as the reactor type, initial compositions, flux level, and burnup. Therefore a reactor may have more than one database describing it, and interpolation on a batch by batch basis of the databases may be necessary for the most accurate description of the reactor and fuel conditions.

For example, the burnup variable can be used to combine on two or more databases. Using the target burnup of the reactor (or more specifically, the batch), the databases will be combined to find the best estimate for a database describing the fuel at the target burnup. The databases are combined using

inverse distance weighing (Shepard's method) as described in [10]. The databases are combined by weighing each one by a value proportional to the databases burnup distance from the target burnup.

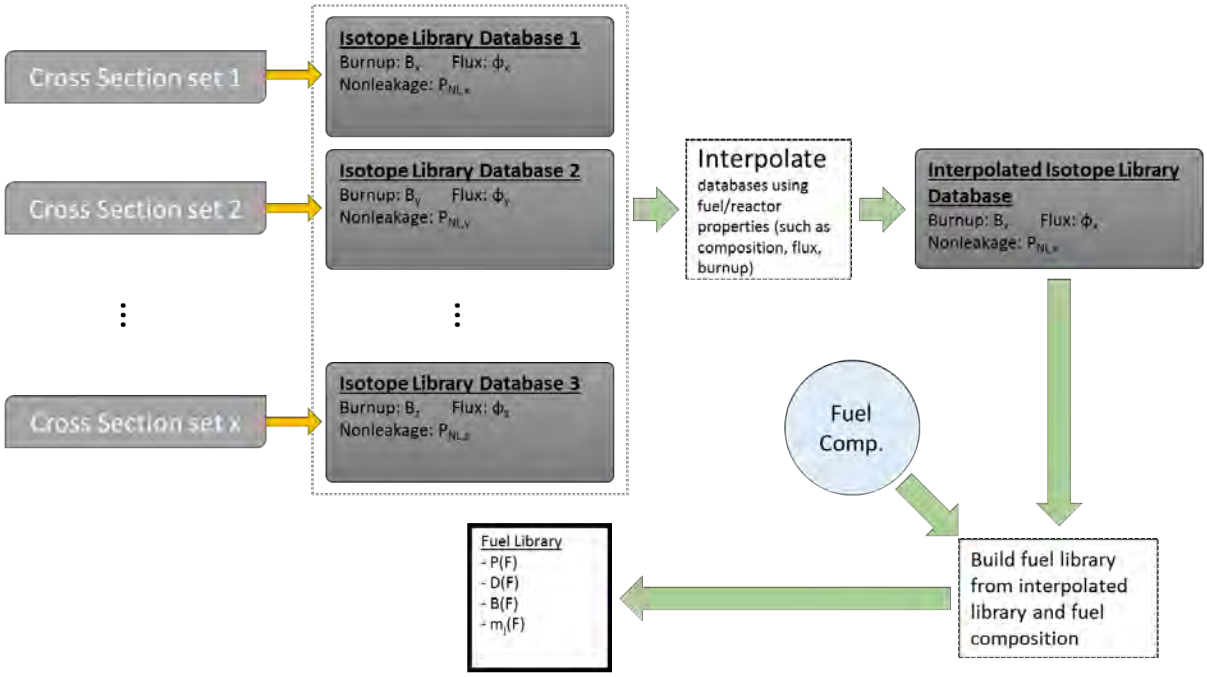

Figure 4 – Fuel Library Generation from known fuel composition.

In order to calculate a fuel library using an isotope library database, a fuel composition needs to be available. This step is straightforward if Bright-lite is operated in the 'forward' mode where cycle and discharge burnups are found for fuel of known composition. However, an iterative approach must be taken in the 'blending' mode, where fuel is created by blending two mass streams with the aim of attaining a multiplication factor of unity when a target burnup is reached.

Examples of mass streams to be blended include U-235 and U-238 (to find the enrichment of uranium-only fuel) and depleted uranium and plutonium of known isotopics (to find the U-to-Pu mixture for MOX fuel). Figure 5 depicts how Bright-lite determines input composition in 'blending' mode. The composition is unknown in this stage and must be guessed by choosing a blending ratio for the two available streams. Next, the resulting burnup from the guessed input composition is calculated (this is also the 'forward' mode) and compared against the burnup target.

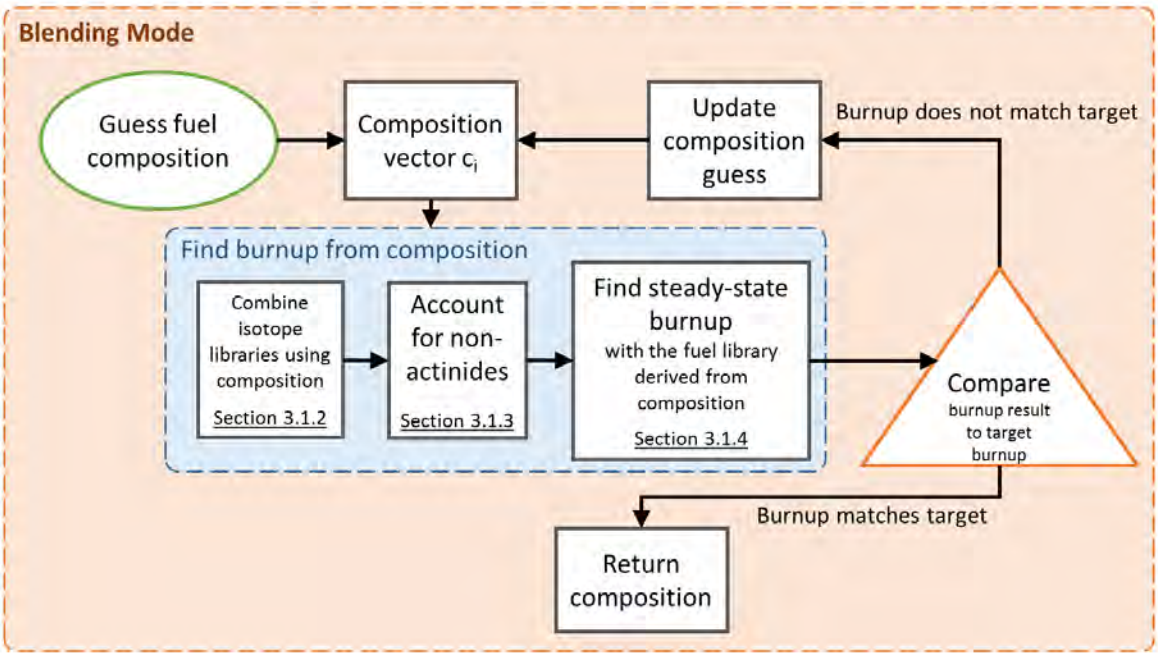

Figure 5 – Input composition finding algorithm of Bright-lite.

In order to describe the operation in 'forward' mode, assume that the fuel composition vector, c_i, is known and the total fuel mass is normalized to unity as described in Eq. (2).

$$\sum_{i=1}^{I} c_i = 1 \quad (2)$$

Hence, c_i is the mass fraction in the fuel of the i^{th} of the I actinide isotopes which are tracked by Bright-lite. Other constituents of the fuel (e.g., oxygen in UO2) are handled with non-fuel materials and discussed in Section 3.1.3. Next, the isotope libraries are combined by mass weighted averaging:

$$P(F) = \sum_{i=1}^{I} (P_i(F) \cdot c_i) \quad (3)$$

$$D(F) = \sum_{i=1}^{I} (D_i(F) \cdot c_i) \quad (4)$$

$$B(F) = \sum_{i=1}^{I} (B_i(F) \cdot c_i) \quad (5)$$

Let the J transmutation and decay product isotopes tracked by Bright-lite be indexed by j. Note that J is greater than I because it includes the parent actinides as well as all of their daughters. Then, the transmutation vector $M_j(F)$ is found as:

$$M_j(F) = \sum_{i=1}^{I} \left(m_{i,j}(F) \cdot c_i\right) \quad \text{for all j} \quad (6)$$

Take fuel with a composition of 96% U238 and 4% U235 as an example. The composition of the fuel is used to combine the isotope libraries to generate the fuel library for this batch. The neutron production rate curve as a function of fluence for this fuel would be found by:

$$P(F) = P_{U235}(F)\cdot 0.04 + P_{U238}(F)\cdot 0.96 \quad (7)$$

Other properties are combined in a similar way using mass weighted averaging. Figure 6 illustrates these properties for the case of the representative PWR loaded with 96% U238 and 4% U235. It can be seen that the neutron production rate eventually drops below the neutron destruction rate. Note that this figure does not consider the presence of control absorber present in the core; while control absorber is certainly present during burnup, the key purpose of this figure is to understand when a multiplication factor of greater than unity cannot be maintained even without control absorber. In this simplified representation, given one batch fuel management the criticality of the core cannot be sustained when the production rate is lower than destruction rate, so the fuel is discharged at this fluence. Note that leakage, spatial effects, and materials that are not actinides or their daughters (e.g., structural materials) have not yet been factored into the calculation.

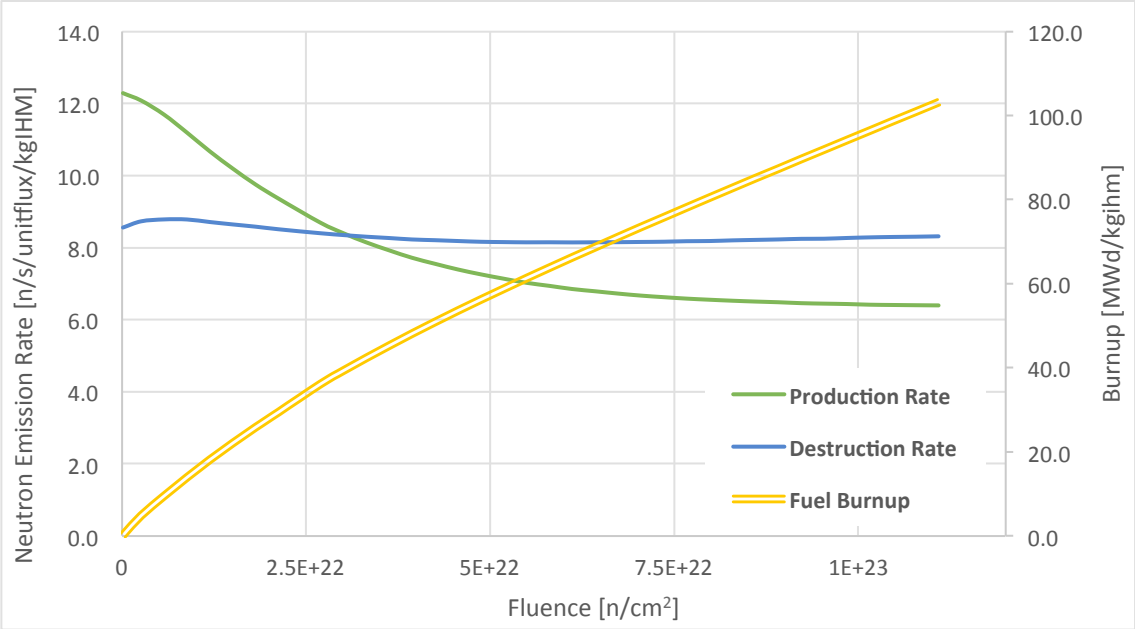

Figure 6 – Example neutron production rate, destruction rate, and burnup for 4% enriched LWR fuel library.

However, fuel is typically separated into batches, and each batch is allowed to have a different starting composition and remain in the core for multiple burnup cycles. In addition, other species aside from the actinides and their daughters affect the neutron balance. Obtaining discharge fluence for a multi-batch core containing non-fuel materials is detailed in the next section.

Contribution of non-fuel materials

The fuel library is generated using the fuel composition. Other materials around the fuel in the core, such as structural materials, also contribute to the neutron economy. The effects of these materials are determined by their composition and cross sections. A fluence-independent neutron production and destruction rate due to these materials is calculated and added to P(F) and D(F) of the fuel to account for their effects.

The mass of non-fuel isotopes present in the core is normalized to a unit mass of fuel, named nf_s. Here, the subscript *s* represents the s^{th} of S non-fuel isotopes and nf is the mass the isotope normalized to a unit mass of fuel. Table 2 shows an example list of non-fuel materials and their normalized masses.

Table 2 – Example list of structural materials and normalized masses for a LWR core.

Isotope *s*	nf_s
H1	7.53E-03
H2	2.26E-06
B-10	8.70E-06
B-11	3.87E-05
O-16	1.18E-01
O-17	4.77E-05
O-18	2.66E-04
Zr-90	1.25E-01
Zr-92	4.25E-02
Zr-94	4.40E-02
Zr-96	7.24E-03
Sn-112	4.49E-05
Sn-114	3.11E-05
Sn-115	1.62E-05
Sn-116	6.97E-04
Sn-117	3.71E-04
Sn-118	1.18E-03
Sn-119	4.22E-04
Sn-120	1.61E-03
Sn-122	2.33E-04
Sn-124	2.97E-04

Next, the cross-section of reactions that absorb and/or emit neutrons are found:

- The (n-gamma) reaction cross-section, $\sigma_{s,\gamma}$ [b]
- The (n-2n) reaction cross-section, $\sigma_{s,2n}$ [b]
- The (n-3n) reaction cross-section, $\sigma_{s,3n}$ [b]
- The (n-alpha) reaction cross section, $\sigma_{s,\alpha}$ [b]
- The (n-p) reaction cross-section, $\sigma_{s,p}$ [b]

The non-fuel material production ($\sigma_{s,P}$) and destruction cross-sections ($\sigma_{s,D}$) are found for every non-fuel isotope.

$$\sigma_{s,P} = 2 \cdot \sigma_{s,2n} + 3 \cdot \sigma_{s,3n} \quad (8)$$

$$\sigma_{s,D} = \sigma_{s,\gamma} + \sigma_{s,2n} + \sigma_{s,3n} + \sigma_{s,\alpha} + \sigma_{s,p} \quad (9)$$

Note that the (n,2n) and (n,3n) reaction cross-sections are accounted once in the destruction cross-section because they cause the destruction of a neutron. The new neutrons these reactions emit are accounted in the production cross-section, keeping the net neutron production correct.

The contribution of the non-fuel (nf) materials to the neutron population is calculated and a total production (p_{nf}) and destruction (d_{nf}) contributions in units of n/s/kg of fuel/unit flux are found as constants independent of fluence. Note that cross-sections are in barns, A_s is mass number of isotope s in kg/mol, and N_A is the Avogadro constant.

$$\mathrm{p_{nf}} = \sum_{\mathrm{s}=1}^{\mathrm{S}} \mathrm{nf_s} \cdot \sigma_{\mathrm{s,P}} \cdot 10^{-24} \cdot \mathrm{N_A} \cdot \mathrm{A_s^{-1}} \quad (10)$$

$$\mathrm{d_{nf}} = \sum_{\mathrm{s}=1}^{\mathrm{S}} \mathrm{nf_s} \cdot \sigma_{\mathrm{s,D}} \cdot 10^{-24} \cdot \mathrm{N_A} \cdot \mathrm{A_s^{-1}} \quad (11)$$

Accounting for these non-fuel effects is explained in the next section.

Thermal Disadvantage Factor Calculation

The previous section formulates the method for determining the neutron production and destruction rates of non-fuel materials. The thermal disadvantage factor is used (for reactors that operate mainly with thermal neutrons) to combine the production and destruction rates of the fuel with those of the non-fuel materials [11].

In order to calculate the thermal disadvantage factor (ξ), thermal utilization is first defined. Since neutrons are not only absorbed by the fuel but the moderator as well, the thermal utilization (*f*) defines the fraction of absorbed neutrons that are absorbed in the fuel.

$$\frac{1}{f} = \frac{\bar{\Sigma}_{\mathrm{a}Mod} \cdot \mathrm{V}_{Mod}}{\bar{\Sigma}_{\mathrm{a}Fuel} \cdot \mathrm{V}_{Fuel}} \mathrm{F} + \mathrm{E} \quad (12)$$

Where *Mod* is the moderator, *Fuel* is the fuel, and *V* is volume. The functions F and E are defined as:

$$F(x) = \frac{x \cdot I_0(x)}{2 \cdot I_1(x)} \quad (13)$$

$$E(y,z) = \frac{z^2 - y^2}{2y}\left(\frac{I_0(y) \cdot K_1(z) + K_0(y) \cdot I_1(z)}{I_1(z) \cdot K_0(y) - K_1(z) \cdot I_1(y)}\right) \quad (14)$$

Where I and K are modified Bessel functions and the variables are defines as:

- $x = a/L_{Fuel}$
- $y = a/L_{Mod}$
- $z = b/L_{Mod}$
- L : diffusion length
- a : radius of the fuel rod

- b : radius of equivalent cell

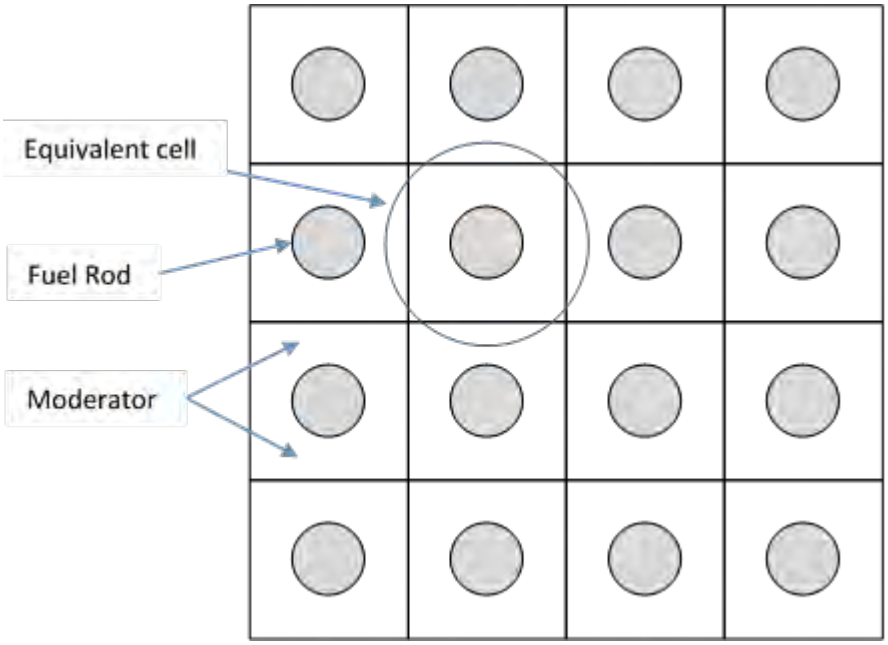

Figure 7 – Equivalent cell for thermal disadvantage calculation

Figure 7 depicts the equivalent cell in a fuel lattice. The macroscopic absorption cross-section determination is detailed in Section 3.2.

Once the thermal utilization is determined, the thermal disadvantage factor (ξ) is found using the formula:

$$f = \frac{\bar{\Sigma}_{aFuel} \cdot V_{Fuel}}{\bar{\Sigma}_{aFuel} \cdot V_{Fuel} + \bar{\Sigma}_{aMod} \cdot V_{Mod} \cdot \xi} \quad (15)$$

Or equivalently:

$$\xi = \frac{\bar{\Sigma}_{aFuel} \cdot V_{Fuel} \cdot (1 - f)}{(\bar{\Sigma}_{aMod} \cdot V_{Mod}) \cdot f} \quad (16)$$

So that:

$$\xi = \frac{\bar{\Phi}_{Mod}}{\bar{\Phi}_{Fuel}} \quad (17)$$

The thermal disadvantage is used to add the non-fuel material contributions to the fuel library. Here, $P_{fuel}(F)$ and $D_{fuel}(F)$ correspond to the production and destruction rates before the non-fuel material contributions are added.

$$P(F) = P_{fuel}(F) + \xi \cdot p_{nf} \quad (18)$$
$$D(F) = D_{fuel}(F) + \xi \cdot p_{nf} \quad (19)$$

Discharge fluence determination

This section describes the methodology for finding discharge fluence in a multi-batch reactor assuming the batch compositions are known. Core criticality is used as the discharge condition by refueling the reactor when the k-value drops below unity. In refueling, the oldest batch is replaced by fresh fuel and the position of the remaining batches is shuffled when the criticality condition is met.

Begin by taking an N-batch reactor. Each batch has a fuel library derived from a reactor database according to the batch composition. Every batch has a fluence $F^{(n)}(t)$ where *n* is the batch index and *t* is the time measured from the beginning of the burnup cycle. Therefore the fluence of the N batches are given as ($F^{(1)}$, $F^{(2)}$, …, $F^{(N)}$) where the fluence of the fresh batch, $F^{(1)}(t=0)$, is zero at the beginning of the cycle.

A time step Δt is defined as the period in which the magnitudes of the average fluxes in each region are assumed constant. This step is selected so that it is much smaller than the cycle length as it will be used to march forward through the cycle by increasing the fluence of every batch. After each time the fluence is incremented, the new criticality of the core will be checked, and the process will be repeated until the core criticality is less than unity.

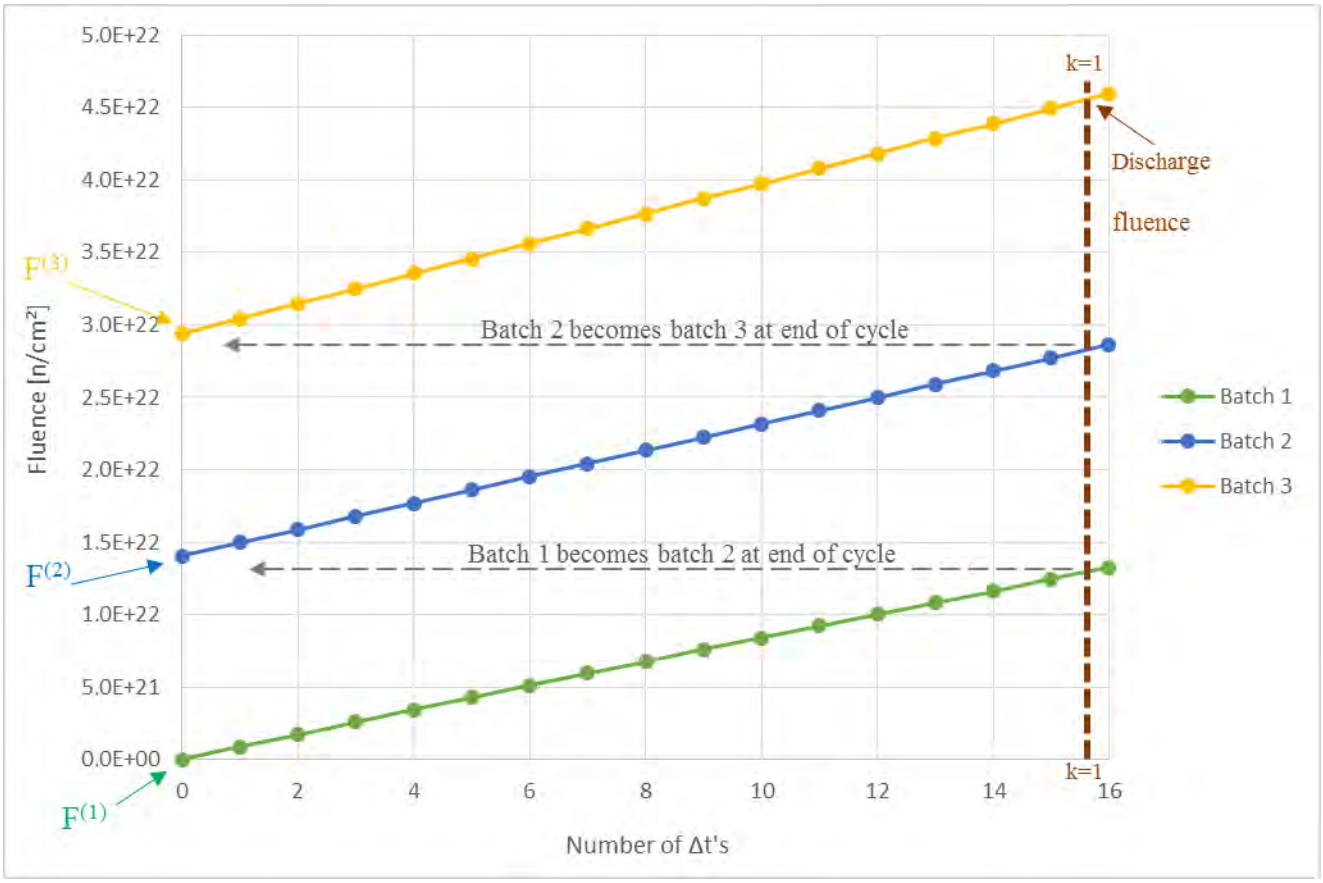

Figure 8 – Discharge fluence determination for an example 3-batch LWR reactor.

Figure 8 depicts the increase in batch fluences marching forward in time for a 3-batch reactor. Initially the first batch has zero fluence, and the third batch has the highest. As fuel is burned the fluence of the batches increase until the criticality of the core drops below unity. This marks the end of the cycle and the oldest batch is discharged.

To march forward through the cycle, Δt is used to calculate $\Delta F^{(n)}(t)$ for each batch and time step. In general, if the volume and time averaged flux ($\Phi^{(n)}(t)$) of batch n over time step starting at t is known, the fluence increment $\Delta F^{(n)}(t)$ can be calculated for the core:

$$\Delta F^{(n)}(t)\ \left[\frac{n}{cm^2}\right] = \Delta t\ [s] \cdot \Phi^{(n)}(t)\ \left[\frac{n}{cm^2} \cdot s\right] \quad (20)$$

Here, it is recognized that the batches do not operate under the same flux. The two methods of finding the relative fluxes of the batches ($r^{(1)}(t)$, …, $r^{(N)}(t)$), termed the equal power share and spatial flux methods, are presented later in this section. These unitless relative fluxes are averaged over the volume occupied by each batch. Depending on the method used to calculate them, they may be fully time dependent and change at each time step, or they may change more rarely – for example, only when the spatial flux distribution is recalculated.

Given that the relative flux of the batches has been found for a time step, a flux scaling factor (with flux units) for the relative fluxes (Φ_{scale}) needs to be calculated. This scaling factor assures that the reactor power output integrated over the time-step is correct. This process is iterative because isotopic transmutation gives rise to a nonlinear relationship between fluence and burnup. During the very first time-step Φ_{scale} is guessed based on the fuel library database, but later the scaling factor from the previous time-step is used as an initial guess (the *g* subscript refers to *guess* values):

$$\Delta F_g^{(n)}(t) = \Delta t \cdot \Phi_{scale,g}(t) \cdot r^{(n)}(t) \quad (21)$$

Next, the change in fluence is used to determine the corresponding change in burnup for the reactor core over that time step. Note that the change in burnup for a given time step is the same for the equal power sharing assumption, but this is not true for the spatial method.

$$\Delta B_g(t) = \frac{1}{m_{core}} \sum_{n=1}^{N} m^{(n)} \left(B\left(F^{(n)} + \Delta F^{(n)}(t)\right) - B\left(F^{(n)}\right)\right) \quad (22)$$

Where $m^{(n)}$ is the mass of batch n. Now the change of burnup ($\Delta B_g(t)$) corresponding to the change in fluence ($\Delta F_g(t)$) is used to determine the power during that time-step for that flux scaling factor:

$$P_{core,g}(t) = \frac{\Delta B_g(t)}{\Delta t} \cdot m_{core} \quad (23)$$

Here the change in burnup ($\Delta B_g(t)$) is in units of MWd/kgIHM, time-step (Δt) is in days, and the mass of the core (m_{core}) is in units of kg. The value for the power of the core from this calculation is compared to the target power to find a flux correction factor (ε_ϕ):

$$\varepsilon_\phi = \frac{P_{core}}{P_{core,g}(t)} \quad (24)$$

$$\Phi_{scale} = \varepsilon_\phi \cdot \Phi_{scale,g} \quad (25)$$

The flux correction factor is used to update Φ_{scale} as given in the equation above. The flux correction factor calculation is repeated until the power ratio of successive iterations, ε_{φ}, falls within a threshold tolerance around unity.

After the completion of this step the fluence of the batches can be updated.

$$\Delta F^{(n)}(t) = \Delta t \cdot \left(\Phi_{scale} \cdot \varepsilon_{\phi}\right) \cdot r^{(n)}(t) \quad (26)$$

$$F^{(n)}(t + \Delta t) = F^{(n)}(t) + \Delta F^{(n)}(t) \quad (27)$$

Next, the neutron production and destruction rate of each batch is found at their new fluence.

$$P^{(n)}(t + \Delta t) = P\left(F^{(n)}(t + \Delta t)\right) \quad (28)$$

$$D^{(n)}(t + \Delta t) = D\left(F^{(n)}(t + \Delta t)\right) \quad (29)$$

The new criticality of the core is calculated. The nonleakage probability (P_{NL}) of the reactor is given in the reactor database. It is used in the criticality calculation as given below. Note that the production (P) and destruction (D) vectors have been updated to include the non-fuel material contributions, weighted by the thermal disadvantage factor.

$$k(t + \Delta t) = P_{NL} \cdot \frac{\sum_{n=1}^{N} P^{(n)}(t + \Delta t) \cdot r^{(n)}(t + \Delta t)}{\sum_{n=1}^{N} D^{(n)}(t + \Delta t) \cdot r^{(n)}(t + \Delta t)} \quad (30)$$

Next, time is advanced by making t into t + Δt. If k remains greater than unity, the simulation continues marching through time and fluence. If k has been found to have dropped below unity the precise fluence at which the cycle ends is calculated using linear interpolation on the k values for two succeeding steps and their corresponding k values. Φ_{scale} is not corrected again for this final step.

Once the discharge fluence of the N'th batch, $F^{(N)}$, is found, its discharge burnup and composition are obtained directly from the batch's fluence-dependent fuel library. Time is reset to zero and the updated fluences of the other batches ($F^{(1)}$, $F^{(2)}$, …, $F^{(N-1)}$) carry to the next cycle with their indices increased by one once this new cycle begins.

Equal Power Share Method

Batches in nuclear reactor cores are not exposed to a constant level of flux throughout their lifetime. Bright-lite incrementally increases the fluence of the batches. Therefore, to accurately increase fluences it is necessary to estimate the relative magnitudes of the average flux of batches. One method to estimate the relative flux is by assuming that every batch contributes equally to the power of the reactor. Batch power density P [MW/kg] can be found using the selected time step Δt [day] and batch burnup during that time step ΔB [MWd/kg].

$$P = \Delta B / \Delta t \quad (31)$$

An example plot showing the dependency between burnup and fluence is given in Figure 9. Recognizing that the curve is non-linear, it can be concluded that in general older batches (with higher fluence) must be exposed to higher flux to keep power density constant.

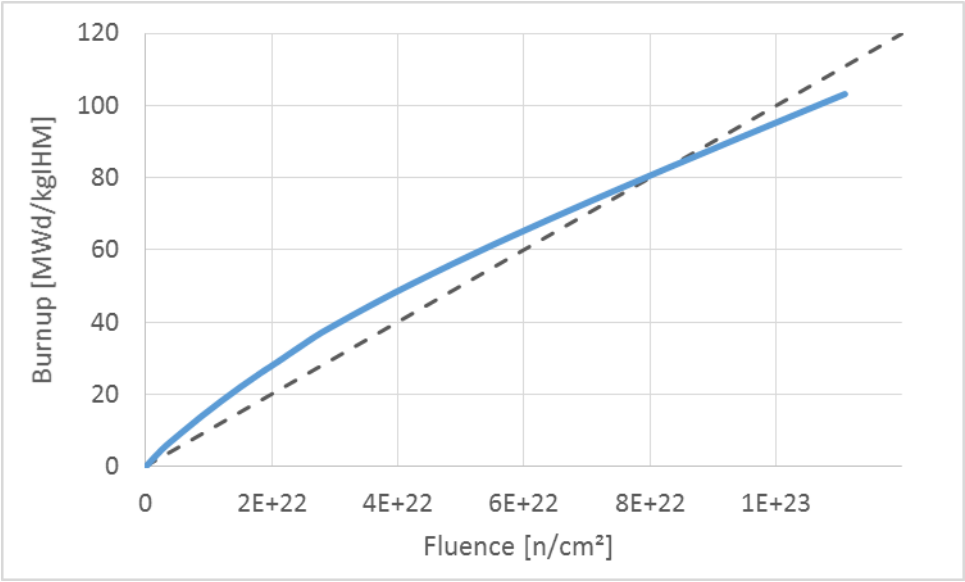

Figure 9 – Increase of burnup with fluence for a 4% enriched LWR fuel.

In order to operate the Bright-lite reactor under the equal power share assumption, every batch much undergo equal burnup. First, the batch with lowest increase in burnup for a given fluence increase (n = N) is selected and its fluence increased. This fluence is found by the following equation, where Φ is the fuel library flux.

$$\Delta F = \Delta t \cdot \Phi \quad (32)$$

The corresponding burnup increase is found next.

$$\Delta B = B\left(F^{(N)} + \Delta F\right) - B\left(F^{(N)}\right) \quad (33)$$

The fluence increase for the remaining batches is found so that the following equation is satisfied.

$$\Delta B = B\left(F^{(n)} + \Delta F^{(n)}\right) - B\left(F^{(n)}\right) \quad (34)$$

This yields a $\Delta F^{(n)}$ for every batch. Since batch *N* was selected to be the batch with the lowest increase in burnup, $\Delta F^{(N)}$ will have the highest value among all *n*. Taking this value as a normalization factor, the relative flux ($r^{(n)}$) of every batch can be determined using the following equation.

$$r^{(n)} = \frac{\Delta F^{(n)}}{\Delta F^{(N)}} \quad (35)$$

This will yield a relative flux of unity for the oldest batch, and relative fluxes lower than unity for the remaining batches. These relative fluxes are then used in the criticality calculation.

Steady-state Burnup Determination

This section describes the determination of the steady-state discharge burnup of fuel batches of a fixed composition. Using the fuel composition (c_i), a corresponding fuel library can be found, called L_{fuel}. The steady-state burnup corresponding to a library L is given by the function BU(L), which returns a steady-

state burnup (b) corresponding to the library L. (This function also requires several reactor parameters saved in the fuel library such as number of batches, non-leakage probability, and tolerance values.)

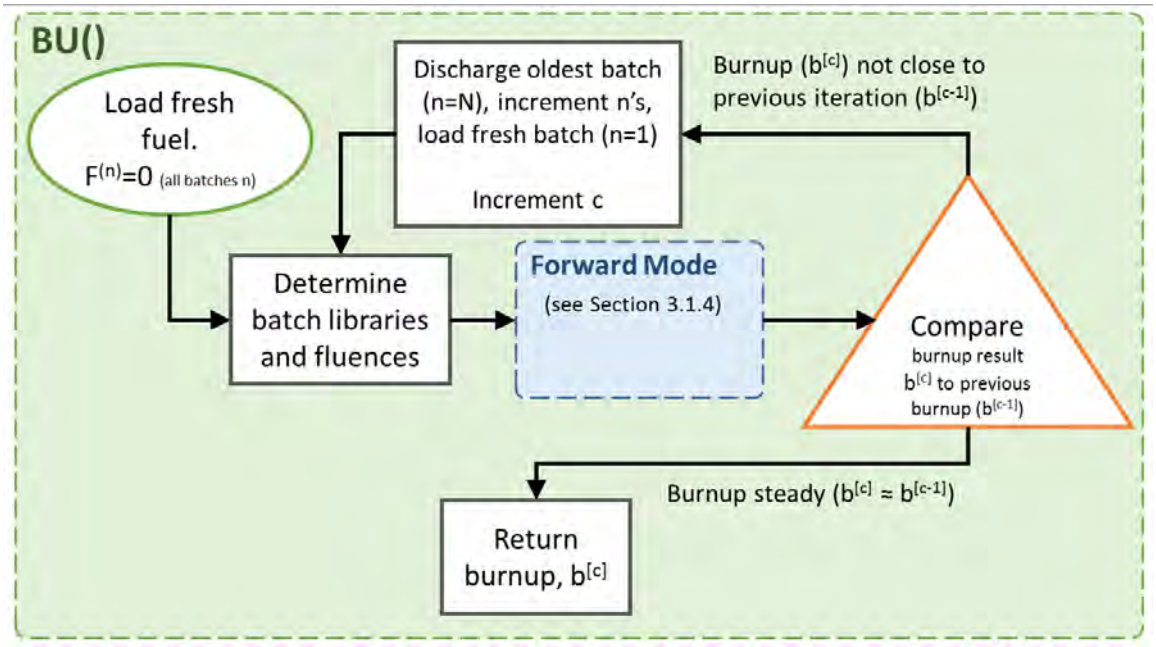

Figure 10 – Operation of BU()

Figure 10 depicts the implementation of BU() for a simple reshuffling scheme. First, the reactor is started with N batches with each batch at a fluence of zero. In this first step the batch numbers (n) from 1 to N are assigned to the batches. Note that the composition of the fuel is already factored in the fuel library that is being used. An irradiation cycle is begun, following the methodology described previously.

Let *c* be the cycle index. Then the first cycle (c=1) fluences are given as $F^{(n),[c]} = 0$ for all n. Once the reactor is loaded the progression of reactor cycles begin. The forward mode calculation within the iteration loop returns the burnup of the oldest batch at the end of cycle *c*, named $b^{[c]}$. In case where more than one batch has the maximum fluence, which only applies to the first startup cycles where initial batches have been resident for equal number of cycles, the batch with highest n (n=N) is selected.

Next, the N^{th} batch is discharged and the batch and cycle indexes are incremented (i.e. the batch index for the new cycle c+1 becomes $F^{(n+1),[c+1]} = F^{(n),[c]}$). A new batch (n = 1) is loaded and assigned zero fluence ($F^{(1),[c+1]} = 0$) and the new core is burned for another cycle. This process can summarized as:

1. Start steady-state burnup determination using the fuel library L.
2. Assign $F^{(n),[1]} = 0$ for all n, where [1] refers to the first cycle.
3. Burn the fuel using forward mode, and obtain the discharge fluence $b^{[c]}$ where [c] is the cycle index. (Forward mode increases the fluence of every batch.)
4. Discharge batch n=*N*, increment all batch indexes by one ($F^{(n+1),[c+1]} = F^{(n),[c]}$), and add fresh batch ($F^{(1),[c+1]} = 0$).

5. Check if $|b^{[c+1]} - b^{[c]}| < \varepsilon$, where ε is a predefined tolerance value. If the difference is within tolerance return $b^{[c+1]}$, otherwise go to step 3.

Note that this calculation process is similar to the one that is carried out when a new reactor is started up, except that the compositions of the initial N batches loaded to the core are identical in steady-state burnup calculation. In reactors with different refueling schedules, the method matches the reactors refueling.

Input fuel calculation

Bright-lite uses an iterative method to blend two fuel streams that will yield a given target burnup. Libraries are created for each fuel stream; a stream can consist of one individual isotope (such as U235 or U238) or mixture of several (such as recycled plutonium or depleted uranium).

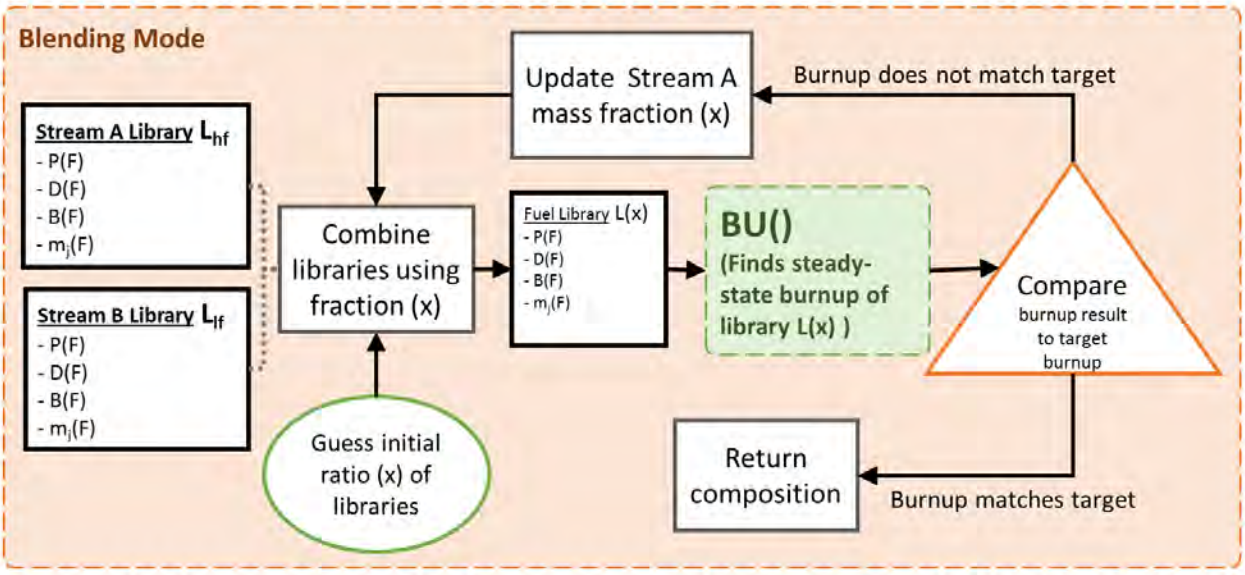

Figure 11 – Operation of 'blending' mode.

Given that two streams are available and their fuel libraries determined, Bright-lite finds the relative mass fraction of each stream so that the combined library will result in the fuel that yields the target burnup (b_{target}). This iterative process, named the 'blending' mode, is depicted in Figure 11. For two fuel streams A and B associated with libraries L_{hf} and L_{lf}, the mass fraction of stream A, x, is used to generate L(x), an input fuel library.

$$L(x) = [L_{hf} \cdot x] + [L_{lf} \cdot (1 - x)] \quad (36)$$

To determine x that yields the target burnup at steady-state, Bright-lite first calculates the steady-date discharge burnup attainable by each stream if it were the only one used. In other words, Bright-lite first

calculates the attainable discharge burnup if x=0, then calculates it for x=1. The calculation for determining the steady-state burnup is performed by the BU(L).

$$\mathrm{BU}(L(x_m)) = \mathrm{b_m} \quad (37)$$

Here the function BU() uses the fuel library L(x) and iterates burnup calculations until an equilibrium cycle is reached and a steady-state burnup, b_m, is found. In some cases, such as a stream of only U238, the core may never achieve criticality. When this occurs, the steady-state discharge burnup is set to zero.

Hence the iterative process for determining the value of x that attains the desired steady-state burnup b_{target} starts by determining the burnups of these two streams, defined as:

$$\mathrm{b_1} = \mathrm{BU}(\mathrm{L_{hf}}) \quad (38)$$

$$\mathrm{b_2} = \mathrm{BU}(\mathrm{L_{lf}}) \quad (39)$$

Where L_{hf} is the library of the first available stream and L_{lf} is the second. Here for b_1, $x_1 = 1$; and for b_2, $x_2 = 0$. Next, x_3 is found by interpolating on b_{target}:

$$x_3 = x_1 + (x_2 - x_1) \cdot \frac{b_{target} - b_1}{b_2 - b_1} \quad (40)$$

Using x_2 and x_3, a secant method approach is taken to find x_{target}. x_{m+1} is found by:

$$x_{m+1} = x_m + \left(b_{target} - \mathrm{BU}(\mathrm{L}(\mathrm{x_m}))\right) \frac{x_m - x_{m-1}}{BU(L(x_m)) - BU(L(x_{m-1}))} \quad (41)$$

New library $L(x_m)$ is calculated using its ratio (x_m).

$$\mathrm{L}(\mathrm{x_m}) = [\mathrm{L_{hf}} \cdot \mathrm{x_m}] + [\mathrm{L_{lf}} \cdot (1 - \mathrm{x_m})] \quad (42)$$

This iteration is repeated until an x value which yields a steady-state burnup close to b_{target} is found, namely so that $|b_{target} - b_m| < \varepsilon$ where ε is the tolerance value (usually 0.001).

Startup Fuel Composition Determination

Reactors have unique startup batch compositions in order to smooth the power profile and more efficiently use fuel during the startup cycles. A fresh core loaded entirely with fuel for steady-state operation would have very high initial cycle lengths as well as uneven power profiles. Therefore it is necessary to determine startup fuel compositions that provide an acceptable cycle length and core power distribution.

Startup fuel compositions in Bright-lite are determined by first obtaining the steady-state fuel composition. This is interpreted as the fuel composition once the startup transient has run its course and the reactor's reloading cycles are reaching equilibrium. In the input fuel calculation described in that section, *x* defines the fraction of two blended fuel streams. In the determination of the startup fuel compositions, the libraries of the two streams that were available for blending are first labeled as L_{hf} and L_{lf}, where L_{hf} is the high fissile stream library and L_{lf} is the library of the stream with low fissile isotope content. The user is tasked with correctly labeling the streams. For example, a natural uranium (or

depleted uranium) stream would be low fissile, whereas an enriched uranium (or reprocessed plutonium) stream is high fissile.

The stream fraction *x* is used to define the fraction of the high fissile stream L_{hf} so that the combined fuel library used in the reactor core is given by:

$$L_{hf} \cdot x + L_{lf} \cdot (1 - x) = L_{fuel} \quad (43)$$

For an *N* batch reactor the fraction *x* is utilized to determine the blending fraction of the first *N* batches for startup. These batches are assigned increasing fissile stream fractions according to the formula (if equal power sharing is used):

$$x^{(n)} = \left(\frac{x}{2}\right) \cdot \left(1 + \frac{n-1}{N}\right) \quad for\ all\ n \qquad (44)$$

Where $x^{(n)}$ refers to the blending fraction of the *n*th stream. Note that there are no batches in the startup reactor core with the fraction *x*. The batch with the steady-state fraction is loaded after the first cycle.

For example, a 3 batch LWR can have a U-235 stream (L_{hf}) and a U-238 stream (L_{lf}) available for blending to determine input fuel. If the steady-state blending fraction is found to be 4.00 %, then the startup blending fractions for the first three batches would be 2.00 % for batch 1, 2.67 % for batch 2, and 3.33 % for the final startup batch. Next, the fuel is burned as described in the section titled Discharge Fluence Determination.

Conclusions

There are many methods for determining the input-output burnup and isotopics of reactors during NFC simulation runs. The accuracy and speed of the simulators are influenced by their selection of method for these calculations. While accuracy is very important for meaningful results, simulation times may be a limiting factor for various types of NFC studies. Therefore, it is important to strike a good balance between runtime of the software and the accuracy of results in order to create a reactor simulator suitable for reactor and fuel cycle studies.

Bright-lite aims to achieve good accuracy with quick execution of the code using the fluence based neutron balance method. This method homogenizes the core to batch-level macro regions to determine the discharge fluence of the reactor batches. However, if the equal power sharing assumption is used to estimate the flux distribution, it has no mechanism to determine the power shares in heterogeneously-fueled reactors where specific power is known to be far from uniform. These more complex cores (such as UOX-MOX mixed, driver-blanket and cores featuring transmutation targets) cannot be simulated by the equal power sharing method since the fluxes of the spatially-dependent batches cannot be known. This limits the applicability and use of Bright-lite, since it cannot be used to simulate many future reactor technologies.

In addition to aiming to achieve good accuracy for specific reactor simulation, Bright-lite also aims to provide users with build in fuel blending capabilities. The methodology used for the fuel blending allows Bright-lite to determine optimized input fuel compositions for a collection of constraints. This feature

allows Bright-lite to handle simulations with changing fuel types, in addition to removing the constraints tied to using prebuilt input and output recipes.

The collection of works presented in this report demonstrate the wide range of capabilities and functionality that has been incorporated into the Bright-lite software. The addition of spatial calculations increases the applicability of the tool to reactor types with uneven power production. On the other hand, the addition of library interpolation capability improves the accuracy of results.

Appendix A.

Citation: R. Flanagan, E. Schneider, C. Bagdatlioglu, "Fuel Composition Calculation Techniques of Nuclear Fuel Cycle Simulators," Physics of Reactors Conference (PHYSOR) 2014, The Westin Miyako, Kyoto, Japan, September 28 – October 3, 2014, on CD-ROM (2014)

Fuel Composition Generation Techniques of Nuclear Fuel Cycle Simulators

Robert R. Flanagan, Erich A Schneider, Cem Bagdatlioglu
Department of Nuclear & Radiological Engineering
University of Texas Austin, Texas, USA
rf8465@utexas.edu
eschneider@mail.utexas.edu
cemb@utexas.edu

ABSTRACT

Nuclear fuel cycle simulators track the flow of materials through the facilities that comprise a nuclear energy system. The composition of these materials, which simulators specify at the elemental or isotopic level, is driven by the neutronic characteristics of the reactors in the system. Therefore, all simulators include a method for generating input and output compositions for the reactor fuel they track, widely known as recipes. This paper surveys the recipe generation approaches taken by five simulators, which range from pre-computed reactor physics modeling to on-the-fly calculations. It concludes with an illustrative example of the canonical parametric recipe generation problem simulators are called upon to solve.

Key Words: **nuclear fuel composition, fuel cycle, cyclus**

1. INTRODUCTION

Nuclear fuel cycle (NFC) simulators track material compositions through facilities such as mines, enrichment, reactor, and waste disposal sites. These simulators are used to evaluate and compare new reactor technologies and fuel cycle strategies by estimating future supply, demand, and flow of materials. Since some reactor technologies rely on material from other facilities as their input, keeping track of fuel compositions as they pass through different reactors is an important problem for simulators to address. This study reviews and compares the methodologies used by NFC simulators for determining fuel composition and burnup. Additionally, one method is demonstrated by an example two stage reactor system where the actinide output of a light water reactor (LWR) is used as the feed to a Canada Deuterium Uranium (CANDU) reactor.

The five NFC simulators reviewed in this paper implement distinct methodologies for addressing the material composition and burnup problem [1]. The chosen NFC simulators are intended to provide a representative survey of available methods.

2. Review of NFC Simulator Reactor Models

The first part of this section presents the review of NFC simulators' approaches to generating reactor fuel input and output compositions. This review focuses on the methodologies implemented by the simulators to determine the fuel isotopics (input/output isotopic composition pairs are often called recipes in the literature) taking into account the transmutation and burnup of the fuel. Characteristics of these simulators are given in Table 1. The second part of the section details three methods that are used by the reviewed simulators.

Table 3. Nuclear recipe generation techniques and software

	Reactor Modeling	Time step	Reactor Deployment	Output Input Recipe Generation	Transport and Burnup Calculation	Tracked Isotopes
VISION	Fleets *Up to 10*	3 months	Look-ahead *User - based on several parameters*	Interpolation (on recipes)	*Pre-calculated* ORIGEN (from WIMS8)	81 Isotopes
NFCSim	Discrete Reactor	Event driven	Look-ahead *(User - constraints and technology preference)*	Interpolation (on cross sections)	*Transport: Pre-calculated Burnup: Calculated on the run* ORIGEN	45 Isotopes
CAFCA	Fleets	1.5 months	Look-ahead *User - fuel technology ratio*	Single input output per reactor type	*Pre-calculated* User Recipe	Elemental
COSI	Discrete Reactor	Event driven	User specified deployment schedules only	Equivalency method	*Calculated on the run : CESAR* Thermal: APOLLO Fast: ERANOS	26 Isotopes
Cyclus/Bright	Discrete Reactor	1 month	Based on demand	Interpolation (on recipes)	*Pre-calculated (burnup using ORIGEN),* Interpolated on the fly	81 Isotopes

2.1. NFC Simulators

2.1.1. VISION

The Verifiable Fuel Cycle Simulation (VISION) code has been developed as a Powersim application by the Idaho National Lab (INL) [3]. It simulates the nuclear fuel cycle by allowing the user to vary energy demand, reactor types, nuclear fuel, and system delays. The software allows the user to input projected energy growth rate and nuclear power market share to determine the characteristics of the fuel cycle.

Neutronics calculations for reactors are performed external to the model. These calculations are saved as recipes which are then used to determine reactor outputs. The recipes used in the software use fixed and discrete input fuel isotopic compositions, which necessitate a shift in the composition vectors over time as the initial input isotopics evolve.

Reactors in VISION need pre-calculated recipes for fresh and spent fuels that correspond to specific type of reactor and fuel. These recipes can be created using any radiation transport and burnup code. Their validity is dependent on the isotopic composition of the fuel initially loaded in the reactor. However, as fuels are recycled in the NFC, their isotopic compositions change. VISION uses a fourth order polynomial fit to correlate between this mismatch of input/output recipe isotopics. The interpolant is one-dimensional and the independent variable is specific to the reactor and fuel type. For LWR UOX type fuel, the correlation is structured as a function of burnup between 33 and 100 MWth-day/kg-iHM, while the independent variable in the fast reactor correlation is the conversion ratio, which can vary between 0.00 and 1.00.

2.1.2. NFCSIM

The Nuclear Fuel Cycle SIMulator (NFCSim) is an event-driven, time-dependent simulation code which models the flow of materials through the nuclear fuel cycle [4]. It is written in Java and uses classes and super-classes for features and functions in

the code.

NFCSim has a criticality and burnup engine for its reactor isotopics and burnup calculations. This engine uses a piecewise-linear, reactor-specific reactivity model. In addition to externally-supplied one group cross sections, several inputs are required from NFCSim for these calculations. They include fuel-to-coolant volume ratio, fuel element geometry, non-fuel material composition, core heavy metal inventory, and target burnup or fuel residence time. The availability of the one group cross section libraries are the limiting factor for the addition of new reactor and fuel types to the software.

NFCSim invokes ORIGEN2 during run time to carry out burnup calculations. The one group cross section libraries are used to track the burnup and neutron production/destruction rates of each actinide (and later its daughters) present in the initial fuel. The reactor recipes are generated for an initial fuel composition by finding a linear combination of production and destruction rates for each constituent. These recipes are constructed per reactor (and per cycle) at the refueling batch level. Additionally, the neutron non-leakage probability is used as a manual calibration tool to match the generated recipes to known benchmark values.

2.1.3. CAFCA

The Code for Advanced Fuel Cycles Assessment (CAFCA) is a fuel cycle simulator developed in VENSIM by Massachusetts Institute of Technology [5]. CAFCA is intended for a large-scale analysis of the nuclear fuel cycle, and has its assumptions geared towards this goal. Time steps are 1.5 months in the software, which does not model distinct facilities. Instead, CAFCA uses a fleet-of-facilities model, only explicitly modeling reactors and reprocessing fleets by tracking the fleets' rates of change. The software requires the user to provide information about the characteristics of reactors such as core mass inventory, cycle length, capacity factor, and composition recipes of fresh and spent fuel for each modeled reactor and fuel type.

CAFCA treats the availability of recycled fuel as the limiting resource in the simulation, with all non-reactor entities set to always meet demand. However, the number of reactors that are fueled by recycled fuel is limited by the supply of fuel recycled from other reactors. The supply of recycled fuel is modeled as ready-to-use fuel inventories (U-TRU or MOX) instead of Pu, RU, or U-TRU inventories. This approach requires the user to define a preferred fuel technology fraction, and the software intrinsically tries to adjust the fuel recycle capacity in order to achieve this ratio. Additionally, all spent fuel is modeled as homogeneous and its decay is not tracked, meaning the code doesn't record the age of discharged fuel [1].

In CAFCA, fuel compositions are specified at the elemental, rather than the isotopic level. Thus charged and discharged fuel is treated as if it were at its equilibrium-cycle elemental composition, and recipes are not adjusted for any changes in the composition of fuel feed stock.. The software tracks minor actinides and plutonium separately for MOX fuels, although in other cases focuses on tracking uranium and TRU in the cycle.

2.1.4. COSI

Commelini-Sicart (COSI) is a nuclear fuel cycle simulator written in Java, developed by the CEA, the French Atomic and Alternative Energies Commission. The user inputs the time period to be simulated as well as the deployment and decommissioning dates of each reactor in the simulation. The reactors in COSI are modeled explicitly, as well as their supporting facilities such as enrichment, separation, and fabrication facilities. The user also provides stock usage preference, i.e. the order spent fuel will be used (older or newer).

COSI uses an equivalence model which takes a reactor and fuel type specific approach to determine the isotopic composition of the input fuel given the isotopic vectors of the feedstocks available to be blended. This model uses a class (type) of fertile material, a class of fissile material, and a ratio of the two classes. For example, a UOX fuel used in thermal reactors is modeled by setting an enrichment of the fuel. On the other hand, MOX fuel is modeled by setting the equivalent plutonium content of the fuel.

Burnup calculations are done separately for thermal and fast reactors in COSI. For thermal reactors, once the equivalence method is used to determine the input fuel composition, a recipe corresponding to this input fuel is generated using APOLLO [6]. APOLLO requires the user to supply core geometries and multi-group cross sections. Fast reactors are modeled in COSI by determining an ideal Pu-239 and U-238 loading of the reactor, and adjusting the input for the deviations from this ideal.

For both thermal and fast reactors, the multi-group cross sections from the transport calculations are collapsed to single-group

values. The subsequent reactor depletion calculations are done using the CESAR code [7]. CESAR is similar to ORIGEN and solves the coupled ordinary differential equations governing nuclide populations due to irradiation and decay over time.

2.1.5. Cyclus/Bright-lite

Bright-lite is a rector isotopics and burnup calculator developed to be a module in the fuel cycle simulator Cyclus [2]. It is developed to solve for output fuel isotopics and burnup given input isotopics. It also performs the inverse calculation of finding the input fuel composition from a given target burnup and isotopic composition of fuel inputs available for blending. Both the burnup and the available fuel inputs (e.g., reprocessed TRU or uranium) are supplied by Cyclus. Cyclus can call Bright-lite to perform a depletion calculation, or to find the required composition when a reactor places an order for a fuel batch.

In its current implementation Bright-lite uses ORIGEN to pre-calculate and parameterize its burnup and transmutation calculations. Libraries are generated for each reactor and fuel type as a function of fluence. These libraries consist of neutron production rates, neutron destruction rates, burnup, and isotopic transmutation vectors for every initial fuel isotope. All decay and transmutation daughters of each initial fuel isotope are implicitly accounted for in the libraries.

For each reactor and fuel type, a transmutation library for every isotope that is allowed to be in fresh fuel is generated. To build a reactor recipe for a given fuel, these tables are combined using a mass-fraction weighted average for each isotope.

2.2. Input Composition Calculations

Every NFC simulator reviewed here has a method for specifying the output composition and burnup if the composition of the input fuel is specified. This section reviews three methods to determine an input composition when multiple feed stocks are available to be blended. The accuracy of these methods is dependent on the number of isotopes tracked as well as the available one- or multi-group cross section data. Their general goal is to determine the input fuel composition that meets user- or fuel cycle simulator-defined output composition or burnup targets.

2.2.1. The reactivity worth approach

This method categorizes isotopes that may be present in a fresh fuel loading as uranium (U) or as transuranic (TRU) material, and is proposed for implementation in VISION [8]. First, k_∞ is defined separately for thermal (*t*) and fast flux (*f*) spectrum by the four factor formula. This formulation is given by equation (45).

$$k_\infty = \varepsilon\, p\, f\, \eta \quad (45)$$

Where the variables multiplied are fast neutron factor (ε), resonance escape probability (p), thermal utilization factor (f), and neutron reproduction factor (η). These variables are also defined for the fast spectrum in equation with the difference arising from usage of fast spectrum cross sections.

Then, the k_∞ for each category is used to find k_{eff} which accounts for reactor losses, given by equation (2) where P_{tNL} and P_{fNL} are the thermal and fast spectrum neutron non-leakage probabilities.

$$k_{eff} = k_\infty P_{tNL} P_{fNL} \quad (2)$$

Later the reactivity is defined by equation (3).

$$\rho = k_{eff,t} - 1 \quad (3)$$

This definition is used to obtain a reactivity balance that allows the initial composition to be adjusted to satisfy a criticality constraint at a desired burnup level. First, every isotope in the input fuel composition (index *j*) is categorized as either U or TRU type. Secondly, these initial isotopes' relative fractions are determined within their category (m^{U_j} or m^{TRU_j}); as well as determining the relative fraction of the two categories (M_U or M_U) using the relative compositions within the category. Using these category fractions as weighing factors to combine the reactivity of each category, an effective reactivity for the specific fuel and reactor is found. Special care needs to be taken when matching the available fuel composition to the domain of known compositions in the multi-dimensional space.

2.2.2. The D-factor method

This method, utilized by COSI, focuses on the net neutron contribution of each initial isotope present in the fuel. First, the neutron consumption per fission of each isotope *j* (D_j) is defined as the "number of neutrons needed to transform the nucleus and its reaction products into fission products" [9]. Therefore a positive *D* corresponds to a net neutron consumption, and a negative *D* to production of neutrons. The D-factor captures the characteristics of each reactor due to its dependence on cross sections, which in turn depend on the flux spectrum of the reactor. Using the input composition of the fuel, the relative fraction of each isotope (m_j) is used to find the D-factor for the fuel (D_{FUEL}) using equation (4).

$$D_{FUEL} = \sum D_j m_j \quad (4)$$

Then, equation (5) is used to determine core criticality.

$$G = S_{ext} - D_{FUEL} - losses \quad (5)$$

Where S_{ext} is a potential external neutron source and *losses* is the neutron losses per fission in the reactor. When *G*, the neutron surplus, is positive, the core is assumed to be critical.

2.2.3. Fluence based neutron balance approach

This method is used by the Bright-lite software and a similar approach was taken in NFCSim. It finds the burnup and transmutation of each possible initial fuel isotope (*j*) while keeping track of the production and destruction of neutrons, all as a function of fluence. These are stored as production (P_j) and destruction (D_j) vectors for each isotope, containing the number of neutrons per initial unit mass as a function of fluence. Using the mass fractions of each isotope in the input fuel (m_j, mass of isotope j per unit mass of fresh fuel) these vectors a combined to create reactor-dependent input vectors (P_{eff} and D_{eff}) using equations (6) and (7).

$$P_{eff} = \sum P_j m_j \quad (6)$$

$$D_{eff} = \sum D_j m_j \quad (7)$$

By dividing the neutron production by the neutron destruction rate after adjusting for leakage, the multiplication factor (k) as a function of fluence is found. The fluence when k-value is equal to unity is used to find the output burnup and isotopics of the reactor. For multi-batch core loading schemes, batches are assumed to share power equally. The discharge fuel burnup and isotopics of the reactor are found by taking the flux-weighted average of the k-values of each batch at their corresponding burnup level.

When calculating the fuel composition needed to achieve a specified target burnup, Bright-lite uses a guess-refining based iteration scheme. In its current implementation, this scheme finds the target composition by interpolating between the two discrete compositions which yield a burnup just above and below the target.

3. BRIGHT-LITE DEMONSTRATION OF METHODOLOGY

A two stage fuel cycle is used to illustrate the Bright-lite. In this cycle, actinides from an LWR provide the feed toa CANDU reactor. This cycle resembles a direct use of spent PWR fuel in CANDU or DUPIC strategy, except that the fission products are removed from the LWR used fuel and the actinides are repackaged into CANDU fuel assemblies.

To illustrate the fluence based recipe calculation process used in Bright-lite, this two stage system was run over a range of initial input enrichments for the LWR. Figure 1 shows the discharge burnup Brigh-light calculates for the LWR given these various initial LWR enrichments. This behavior matches the values produced by the VISION curve fit for its LWR.

The initial enrichment-dependent output isotopics from the LWR affect the burn up of the DUPIC reactor system. This behavior is due to the shifting concentrations of fissile, fertile and neutron poisons in the LWR spent fuel, as seen in Figure 2.

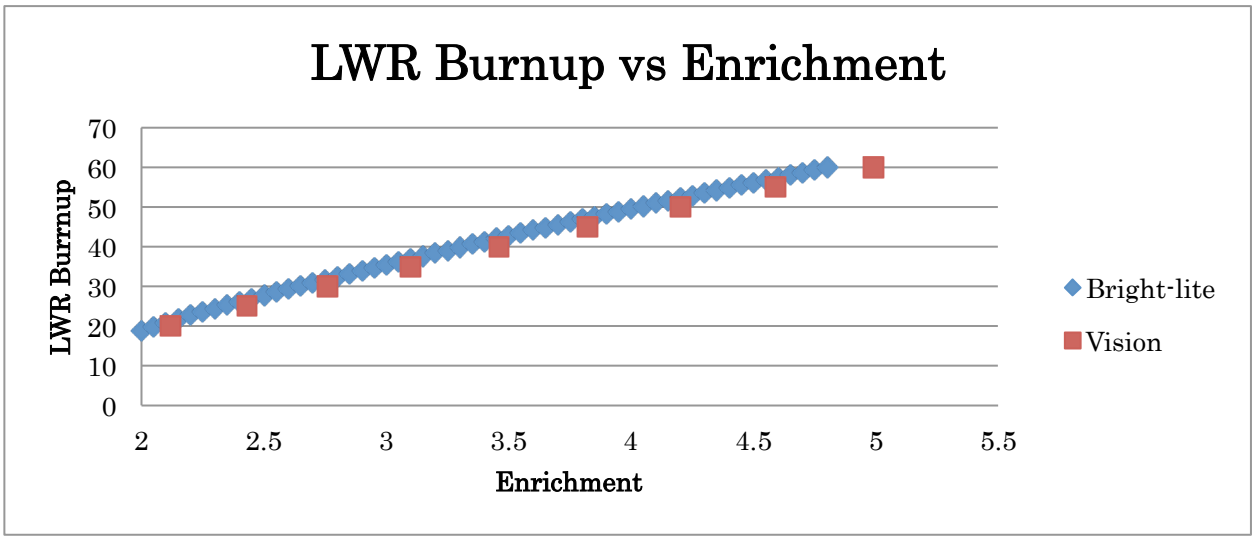

Figure 12. The effects of enrichment on the burn up of the LWR

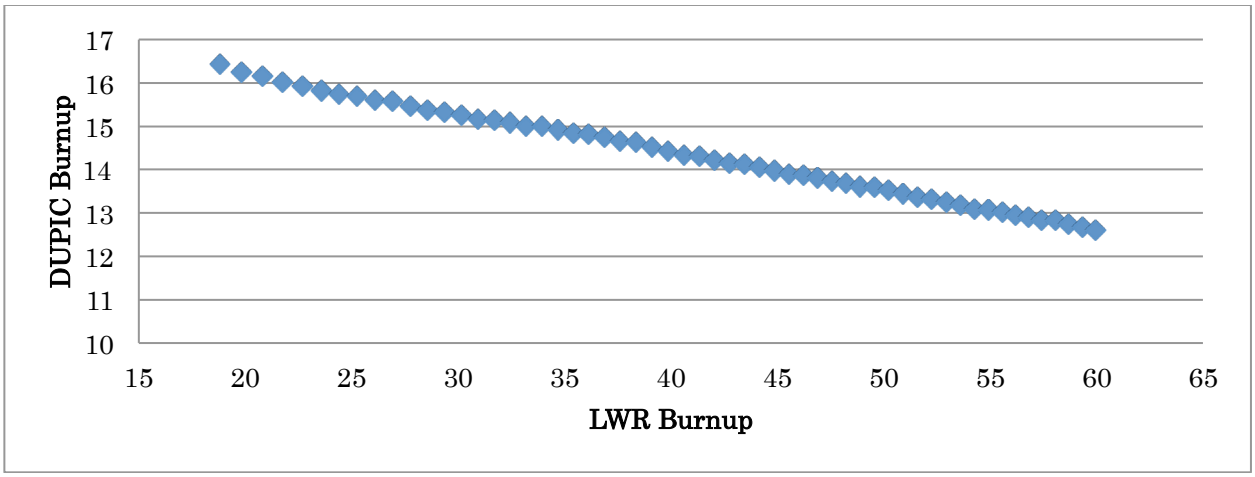

Figure 2. Burnup behavior of the DUPIC reactor given the burnup of the LWR.

For example, Figure 3 shows that the concentration of U236 in LWR spent fuel increases rapidly with burnup, while the U235 concentration declines and the Pu239 remains approximately level. Hence the discharge burnup achievable in the DUPIC reactor, which deteriorates as the LWR burnup increases (Fig. 2), is driven by the increased presence of neutron poisons (specifically U-236) in the actinides.

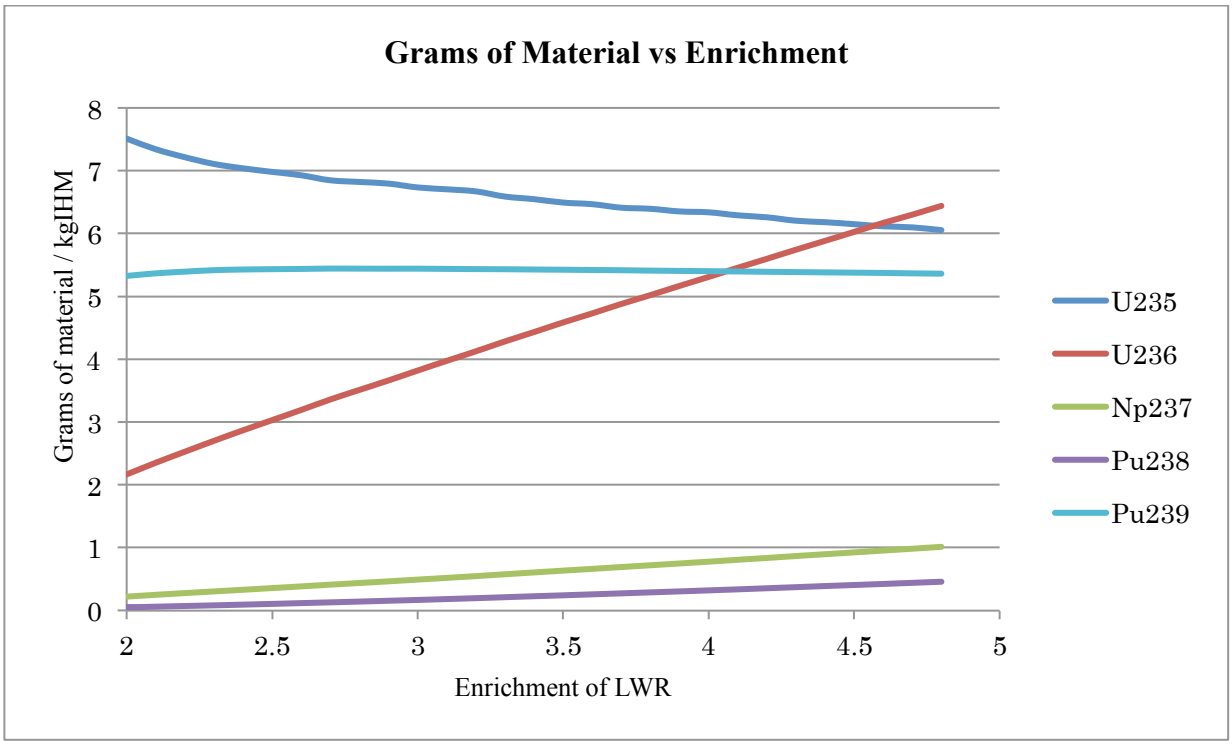

Figure 3. Output isotopics of the LWR by enrichment of the LWR

These results demonstrate that the Bright-lite approach can generate scenario-specific material balances for a coupled reactor system where recipes generated for one reactor or fuel type explicitly affect a downstream recipe. This allows for a fuel cycle simulator to model the start-up, decommissioning and evolving material flows between individual reactors through time within a reactor fleet.

5. CONCLUSIONS

Fuel cycle simulators use various methods for determining the input and output isotopic recipes of the nuclear reactors they simulate. When simulating the evolution of a nuclear reactor fleet over time, it is advantageous to adopt an approach that is capable of generating input output composition recipes during simulation runtime. The Bright-lite implementation of the LWR DUPIC cycle illustrates one approach for explicitly handling the coupling of the material balances between two reactor types.

REFERENCES

[1] M. Gidden, "An Agent-Based Modeling Framework and Application for the Generic Nuclear Fuel Cycle," 2013.

[2] CNERG Fuel Cycle Group, "Cyclus," 01 01 2014. [Online]. Available: http://fuelcycle.org/. [Accessed 01 01 2014].

[3] J. J. Jacobson, G. E. Matthern, S. J. Piet and D. E. Shropshire, "VISION: Verifiable Fuel Cycle Simulation Model," in *ANFM*, 2009.

[4] E. A. Schneider, C. G. Bathke and M. R. James, "NFCSIM: A Dynamic Fuel Burnup and Fuel Cycle Simulation Tool," *Nuclear Technology,* vol. 151, 2005.

[5] L. Guerin, "Impact of Alternative Nuclear Fuel Cycle Options on Infrastructure and Fuel Requirements, Actinide and Waste Inventories, and Economics.," MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2009.

[6] R. Sanchez, I. Zmijarevic, M. Coste, E. Masielo, S. Santandrea, E. Martinolli, L. Villate, N. Schwartz and N. Guler, "APOLLO2 YEAR 2010," 2010.

[7] J. Grouiller, J. Vidal, A. Launay, Y. Berthion, A. Marc and H. Toubon, "CESAR: A Code for Nuclear Fuel and Waste Characterisation," in *WM*, Tucson, AZ, 2006.

[8] S. Yee, "Reactivity Worth Approach".
[9] OECD, "Physics and Safety of Transmutation Systems," Nuclear Energy Agency, 2006.
[10] M. Bell, "ORIGEN: the ORNL isotope generation and depletion code," Oak Ridge National Lab, Tenn. (USA), 1973.

Appendix B.

Citation: C. Bagdatlioglu, E. Schneider, R. Flanagan, “Using Spatial Flux Calculations to Improve the Fluence-Based Neutron Balance Approach”, Presented 2015 ANS Annual Meeting.

Using Spatial Flux Calculations to Improve the Fluence-Based Neutron Balance Approach

Cem Bagdatlioglu, Erich Schneider, Robert Flanagan.

Department of Nuclear & Radiological Engineering University of Texas Austin, Texas, USA
cemb@utexas.edu
eschneider@mail.utexas.edu
rf8465@utexas.edu

INTRODUCTION

Nuclear fuel cycle (NFC) simulators are used for the evaluation and comparison of new fuel cycle technologies. In order to carry out this comparison, it is necessary to determine with sufficient accuracy the various isotopic compositions of materials passing through the fuel cycle facilities. In some cases, it is very useful to determine the isotopic compositions of materials passing through the fuel cycle based on certain parameters such as reactor burnups or reactor conversion ratios.

One method to calculate these properties is the fluence-based neutron balance approach. The method was previously used and tested in the event-driven simulation code NFCSim [1]. It is also currently being used in a Cyclus (a NFC simulator developed at UW Madison [2]) reactor module known as Bright-lite [3].

This method is limited to only one fuel type and is only accurate in an approximate sense even for uniform fuel loadings. The improvement to this method defined in this paper uses macro region based coupled spatial flux and burnup calculations to enable the fluence-based approach to treat heterogeneous cores and fuel loadings. Homogeneous regions are defined within the reactor core at both the micro (fuel-cladding regions) and macro (batches) levels. Each region has a unique, time-dependent relative flux, which is used to advance the fluence of the material(s) within that region.

The introduction of regions improves the treatment of batch-level homogeneous problems and enables the user to model heterogeneous cores. For example, a breeder reactor cannot be modeled with the simpler method because a breeder reactor uses a power producing region and a breeding region. The new method enables the modeling of this type of reactor as well as improving results from mixed fuel types (such as uranium and plutonium in light water reactors). The method can also be used to determine the effects of micro heterogeneities such as the disadvantage factor.

COUPLING SPATIAL FLUX CALCULATIONS

The fluence-based neutron balance approach can simulate multiple batches by separately tracking the fluence of each batch. Even though older batches will contribute less to the neutron economy within the core, the method cannot directly determine the neutron exposure of different batches. In addition, in the case where the batches are materially very different from one another (such as the presence of a breeding blanket in a breeder reactor core), correctly propagating fluence is not possible.

The spatial flux calculations are used to eliminate this drawback by introducing a spatial geometry calculation to determine the relative flux of each batch. These fluxes are then used to scale the increase of fluence more accurately.

First the batches are organized in concentric circles as shown in Fig. 1. The arrangement of the batches can vary depending on the problem. For example, in a breeding reactor the breeding blanket would be located at the outermost region to maximize neutron utilization. For light water reactors, the outermost region can be set as a sufficiently thick moderator.

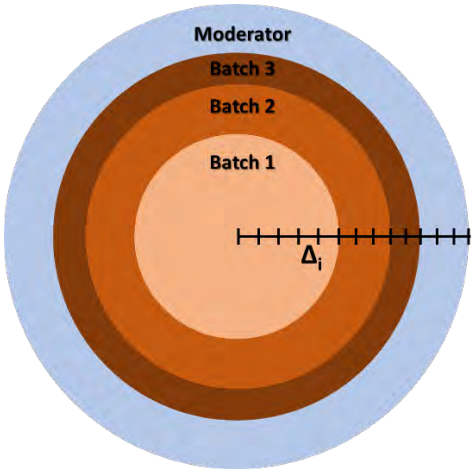

Fig. 1. Spatial batch arrangement for flux calculation.

In order to determine the relative fluxes of the multiplying media, an infinite concentric cylinders problem is solved using finite differencing. For the solution the following space and material property dependent discretized diffusion equation is used.

$$-\frac{D}{\Delta^2}\left(\phi_{i+1}\left(\frac{2i+1}{2i}\right)-2\phi_i+\phi_{i-1}\left(\frac{2i-1}{2i}\right)\right)+\left(\Sigma_a-\frac{1}{k}\nu\Sigma_f\right)\phi_i=0 \qquad (1)$$

Here, the subscript *i* refers to the discrete cells with radial thickness Δ, *D* is the diffusion coefficient, Σ_f is the macroscopic fission cross section, Σ_a is the macroscopic absorption cross section, and φ_i is the flux on cell *i*. The criticality of the core, k, is taken to be equal to unity in an operating reactor. The matrix equivalent of this equation used in the problem solution is given in Eq (2).

$$\underline{\underline{A}}\underline{\phi}-\frac{\nu\Sigma_f}{k}\underline{\phi}=\underline{0} \qquad (2)$$

The absorption and fission cross sections of each region are derived from the material properties described by the fluence-based data. The transport cross section, used to determine *D*, is kept independent of fluence.

Since the mass, and therefore the area, of each batch is kept equal, in a cylindrical geometry the innermost batch will include more cells than other batches. The relative flux of a batch is determined by averaging the flux of each cell within it by using the cylindrical area of the cell as its weight. These relative fluxes are then used to scale the added fluence of each batch. The scaling is done so that the reactor power production remains constant.

Cross sections continually change with increasing fluence, which in turn affects the relative flux. Therefore, the spatial flux calculation needs to be repeated for every fluence step.

PRELIMINARY RESULTS

In order to estimate the relative flux of batches without spatial calculations, the assumption that every batch is producing equal power can be made. Using this assumption and the inferred fission cross section of each batch, relative fluxes can be calculated. This method for estimating the relative flux is used in Bright-lite (Cyclus).

Relative flux data using the equal power sharing assumption was collected in Bright-lite for two reactor cases. The same cases were then used to calculate the relative flux with the spatial flux method. Fig. 2 shows the first case, where a 3-batch light water reactor with 3.5% enriched uranium oxide fuel was used. The innermost region, batch 1, is normalized to one in both methods.

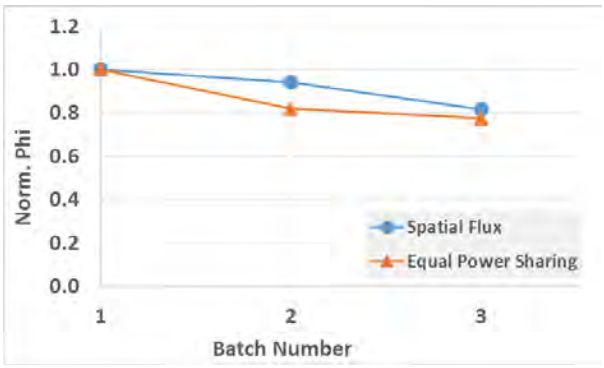

Fig. 2. Relative flux comparison for a 3 batch core.

Fig. 3 shows a light water reactor with 5 batches with 4% enriched uranium fuel. In both cases the outermost moderator region flux is omitted from results since the fluence of moderators are usually not tracked.

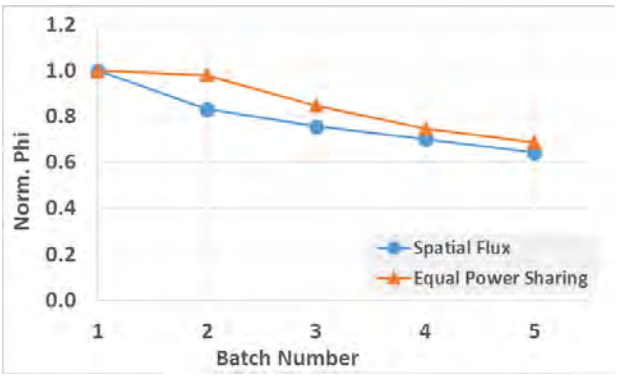

Fig. 3. Relative flux comparison for a 5 batch core.

FUTURE WORK

The implementation of the spatial flux calculations within Bright-lite is underway. Once completed, cases which previously could not be solved using the fluence-based neutron balance method alone will be calculated, such as a fast reactor case with a breeding blanket. The effects of the improved fluence scaling will be evaluated for other reactor types such light water reactors and mixed fuel cases.

The introduction of regions enables the possibility of arranging the fuel in other arrangements as well. Slab geometries, for example, can be used to estimate the fuel disadvantage factor to improve results. More complex geometries can be used for increased accuracy in various fuel arrangements when there are multiple fuel types.

ACKNOWLEDGMENTS

The authors would like to thank Anthony Scopatz and the Cyclus team.

REFERENCES

1. E. A. SCHNEIDER, C. G. BATHKE AND M. R. JAMES, "NFCSIM: A Dynamic Fuel Burnup and Fuel Cycle Simulation Tool," *Nuclear Technology,* vol. 151, 2005.

2. CNERG Fuel Cycle Group, "Cyclus," 01 01 2014. [Online]. Available: http://fuelcycle.org/. [Accessed 01 01 2014].

3. R. R. FLANAGAN, E. A. SCHNEIDER, C. BAGDATLIOGLU, "Fuel Composition Calculation Techniques of Nuclear Fuel Cycle Simulators," *PHYSOR 2014, The Westin Miyako, Kyoto, Japan, September 28 – October 3, 2014.* CD-ROM (2014)

Appendix C.

Citation: R. Flanagan, E. Schneider, C. Bagdatlioglu, "Multidimensional Cross Section Library Interpolation for the Bright-lite Reactor Modeling Software," Global Fuel Cycle Conference, 2015.

Multidimensional Cross Section Library Interpolation for the Bright-lite Reactor Modeling Software

Robert Flanagan, Erich Schneider, Cem Bagdatlioglu
University of Texas Austin
Austin, TX
Email: rf8465@utexas.edu,

Abstract –
The Bright-lite reactor modeling software is a medium fidelity tool that uses cross section libraries to generate burnup and isotopic composition vectors. Using these curves the software is capable of determining output compositions and burnups for a given input fuel composition. This dependence on cross section libraries limits the effectiveness of Bright-lite to specific reactor designs. To increase the accuracy of the tool past these specific libraries a multidimensional interpolation method is used to generate interpolated libraries. The interpolation method used is derived from the Shepard multidimensional interpolation method. This method was chosen because it does not require a gridding structure for the libraries used to perform the interpolation. Additionally this method allows for interpolation over the entire n-dimensional space that a reactor design might be modeled with. This paper demonstrates the effectiveness of the interpolation method to interpolation between the libraries used by Bright-lite. A pre-generated set of libraries are used to interpolate on the parameters of burnup and enrichment. Interpolations will first be performed to show the interpolation method is capable of matching known libraries within the bounds of the interpolation. In addition, a test will be conducted to show the effectiveness of the interpolation method on systems of libraries that do not create a grid for bounding the parametric system. This technique will allow for the rapid simulation of several reactor models using unique burnup and enrichment characteristics.

I. INTRODUCTION

The diversity of possible nuclear fuel cycles using present and future technologies makes medium fidelity reactor models very attractive for fuel cycle simulation. Medium fidelity models allow the system to achieve meaningful results in a short amount of time. This time saving aspect is important for performing parametric studies on specific fuel cycles, as the number of parameters to sweep through in a given fuel cycle (and even for a given reactor in the fuel cycle) can be quite large.

Additionally, in advanced fuel cycles the isotopic composition of input and output fuels may change over time. Reactor startup offers one example input fuel compositions that vary between successive batches. Other examples include operations where enrichment and discharge burnup are gradually being increased and reloads of recycling reactors with time varying sources of feed stock.

To accomplish the goal of quickly and flexibly being able to blend feed stocks and calculate fuel burnup for a wide range of reactor types, a new reactor burnup simulation module, Bright-lite [1], being developed for the Cyclus [2] fuel cycle simulator, uses prebuilt criticality, burnup, transmutation matrices to calculate the input and output isotopic composition. The method used by this module is fast and computationally inexpensive. Therefore a large number of reactor technologies can be investigated quickly.

Bright-lite is capable of operating in two different modes. In *forward mode* fuel of known composition is passed directly to the reactor and burned. This method will return the burnup of the passed fuel at discharge as well as the isotopic composition of the spent fuel. In *blending mode* Bright-lite has multiple feed stocks available for blending. Given a target burnup, it computes the appropriate blend of feed stock streams, generating a fresh and corresponding spent fuel composition that will achieve that burnup.

Libraries used in Bright-lite are specific to characteristics of a reactor and its fuel. This means there might exist a light water reactor library that was generated from 3% enriched fuel with a target burnup of 33MWd/kgIHM (3%-33MWd), and another library using the same reactor design however the fuel was enriched to 5% and a target burnup of 60MWd/kgIHM (5%-60MWd).

Each library is composed of four main components; neutron production (n/s/flux/kg), neutron destruction (n/s/flux/kg), burnup (MWd/kgIHM), and a transmutation matrix. Each of these components is a function of fluence.

The units on neutron production and destruction, n/s/flux/kg, are those used in ORIGEN2.2 [3]. This unit is a representation of the amount of neutrons being produced by a given composition of fuel per second, per mass, per unit flux. When building the Bright-lite libraries a kilogram of fuel is used and therefore the unit reduces to n/s/flux.

A unique library is needed for each set of enrichment and burnup values because changing these values changes the neutron spectrum of the reactor by changing the composition of the material in the core. This affects the one group cross sections that are used to generate the Bright-lite libraries and changes the behavior of the reactors criticality with fluence.

These effects make simulating reactors outside the scope of the pre-generated libraries inaccurate to varying degrees. For example trying to simulate a reactor of 2.2% enrichment and a burnup of 25MWd/kgIHM using the 3%-33MWd could result in much different output fuel compositions than if a 2.2%-25MWd library were used to conduct the simulation. If Bright-lite is operating in forward mode passing the same fuel enrichments to these two libraries will result in different burnups. Conversely, passing the same burnup target to two libraries in blending mode will result in two different fresh fuel compositions.

The libraries used in Bright-lite are time consuming to generate. These libraries require running a Monte Carlo simulation that will determine one group cross sections for a given reactor design. To reduce the need to externally generate new libraries through Monte Carlo simulations a method for library interpolation is being developed to provide increased fidelity by dynamically creating libraries interpolated from existing libraries. The interpolation method is based on an inverse distance interpolation technique. This method allows the interpolation scheme to operate effectively in non-uniform grids over many different variables: burnup, fuel composition, fuel pitch size, etc.

II. METHOD

One method to increase the fidelity of Bright-lite would be to create libraries for all reactors with the values each of their parameters could take. This method would be computationally expensive because the list of parameters per reactor can be quite large; pitch size, thermal power, enrichment (or fuel composition if more than two fuel isotopes are present), burnup, pin size, reactor temperature, cladding type, moderator type, geometric dimensions of the reactor, etc.

Consider generating libraries for two such parameters with M values; for example burnup at values of; 30, 40, 50, 60, 70, 80, 90, and 100 MWd/kgIHM (M = 8) and 8 evenly spaced enrichments between 3 and 7% U235. The total number of combinations of parameters, and hence the number of libraries needed, N, scales as follows:

$$N = \prod_{p=0}^{P} M_p \qquad 1$$

Where
p = a parameter
P = total number of parameters
M_p = number of values associated with the p^{th} parameter.

TABLE I

The scaling for libraries required by parameter

Parameters	Values	Libraries Required
1	8	8
2	8	64
3	8	512
4	8	4096
2	4	16
2	8	64
2	16	256
2	32	1024

Table 1 shows how quickly the number of libraries required to describe a system of parameters becomes untenable to generate.

A less computationally expensive technique for increasing the fidelity of Bright-lite is the use of a multivariate interpolation method. Interpolation allows for the creation of dynamic libraries to be used by Bright-lite to simulate sets of reactor parameters which do not have an existing library associated with them. Some interpolation capability would always be necessary unless Bright-lite restricted use cases to only conditions that exactly corresponded to those of an existing library. The objective of this work is to implement an interpolation method that efficiently and accurately makes use of existing libraries to the fullest extent possible when generating new ones.

Interpolating on Bright-lite libraries poses several challenges. The first is that the interpolation must be multivariate. Each variable that is being interpolated upon might have its own scale and units associated with it. Therefore the interpolation scheme must be able to normalize the distance to the same scale for each variable.

To solve this problem all variables are normalized to a new scale from zero to one ([0, 1]). This is done by subtracting the minimum value from all values in a variable's range, and then dividing all of the resulting values by the total range the variable encompasses.

$$v_{i,new} = \frac{(v_i - v_{min})}{(v_{max} - v_{min})} \qquad 2$$

Where
i = the i^{th} parameter
v = value of the variable

This normalization not only converts the scale of every parameter to a common range but also makes each parameter dimensionless.

II.A. Shepard's Method

The second problem is that the libraries in Bright-lite will not necessarily be evenly gridded and therefore the libraries will not form an even multidimensional grid of points. In fact, due to the combinatoric issues raised earlier it would be unwise to aim for creating such a regular, structured grid as the number of dimensions to be interpolated upon grows large. Hence Bright-lite must not rely on an interpolation scheme that requires a grid system to operate.

This issue can be overcome using an inverse distance interpolation scheme. Bright-lite uses a modified version of Shepherd's Method [4] to perform the interpolations on the libraries.

Since all variables are rescaled to range from zero to one, the inverse distance method allows for distance comparisons between all of the variables no matter their units. An important strength of this method is that it makes use of all existing libraries, not just the nearest neighbors. An adjustable weighting function allows the modeler to tune the influence of adjacent relative to far-away libraries. In a departure from the classical implementation of the method, the weighting will here also allow different rules to be applied when libraries are being extrapolated rather than interpolated.

The key step of a Shepherd's Method calculation is to perform the inverse distance calculation to determine the new values for the newly created library. The distance weighting factor is determined using the following equation.

$$p_j^{\alpha} = \left|P - P_j\right|^{\alpha} = \sqrt{(x - x_j)^{\alpha}} \qquad 3$$

Where:
P = the target value of the interpolation for the parameter P
P_j = the value of the j^{th} library used in the interpolation for the parameter P
α = a smoothing factor for the interpolation.

The distance weighting factor is then used in the following formula to perform the interpolation.

$$U(P) = \frac{\left[\sum_{i=1}^{N} F_i * \prod_{j \neq i} p_j^{\alpha}(P_j, P)\right]}{\sum_{i=1}^{N} \prod_{j \neq i} p_j^{\alpha}(P_j, P)} \text{[5]} \qquad 4$$

Where:
U(P) is a function for the interpolated value at point P
N = number of known libraries
F_i = value of data point in library i.

Equation 2 is applied to each member of the existing Bright-lite data libraries to generate a new custom library with identical structure. Bright-lite then stores this library in memory to be used for the rest of the simulation.

The α factor gives the interpolation some flexibility in how it behaves. Consider a 2-dimensional system with three points used to perform an interpolation. The three arbitrarily chosen points are marked in red in the examples shown in Figures 1, 2, 3 and 4.

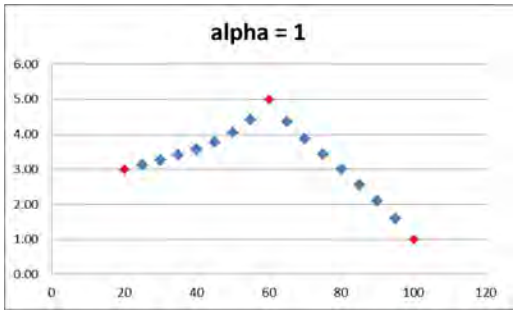

Figure 13 Two-dimensional interpolation, alpha = 1

Even at a=1, Shepherd's method does not reduce to a linear interpolation between nearest neighbors because there are three points, and therefore each library has some impact on interpolation. This is why there is slight bowing between the first and second points. Now consider higher values for α, see Figs. 2 through 4.

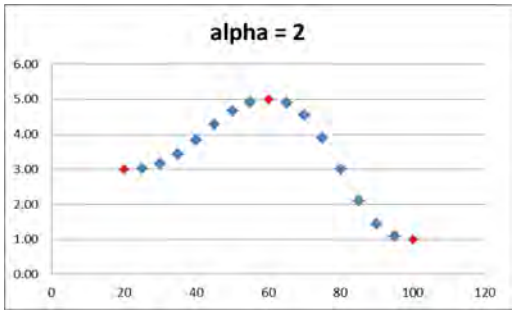

Figure 14 Two dimensional interpolation, alpha = 2

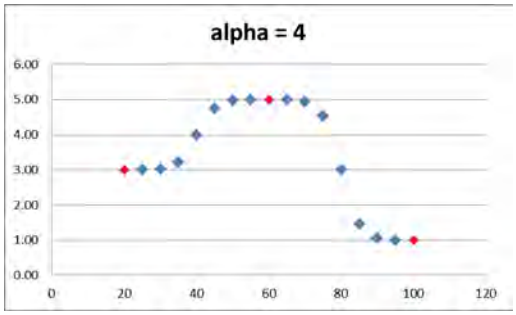

Figure 15 Two dimensional interpolation alpha = 4

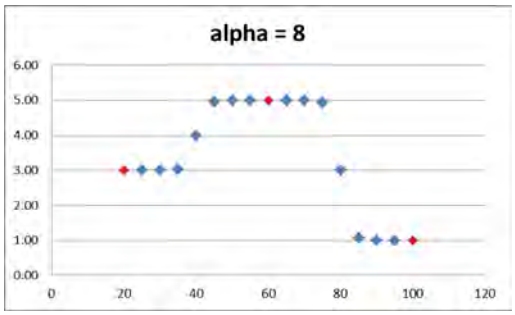

Figure 16 Two dimensional interpolation alpha = 8

As α increases, the weighting of the nearest point becomes stronger. As can be seen in figure 4, once α becomes sufficiently large the method simply enforces the value of the closest known data point for the interpolation. In future work the sensitivity of the interpolation and Bright-lite results to alpha value will be more fully quantified.

One additional challenge this method must overcome is extrapolation of points outside the bounds of the interpolating parameter space. To handle this case, if some but not all of the parameters are being extrapolated the smoothing factor α is increased to 4. This drives the weighting of the system to favor the closest valued library for those parameters. If the system determines that all parameter values chosen as interpolation values are actually outside the parametric bounds of the libraries, the closest library to where the new dynamic library was requested will be used. This is achieved by setting the value of alpha in equation 2 to be very large. Larger alpha values weight nearby points more heavily in Shepherd's scheme.

II.B. Blending Mode Reactor Cases

Testing the interpolation method using the blending mode of operation is conducted with three libraries. The libraries have the following characteristics.

1. Low LWR
    a. Enrichment: 2.2% U235
    b. Burnup: 20MWd/kgIHM
2. Standard LWR
    a. Enrichment: 3% U235
    b. Burnup: 33MWd/kgIHM
3. High LWR
    a. Enrichment: 5% U235
    b. Burnup: 50MWd/kgIHM

All schemes assume 3 batch fuel management. The first test will use the two bounding libraries (Low and High) to create a dynamic library. The objective of the test is to show that the dynamically generated library data calculates the correct fresh fuel enrichment to attain the targeted burnup value. The enrichment and discharge isotopics calculated by Bright-lite using the dynamic library will be compared against those obtained if the Standard library is used. The Standard library was not created by interpolation but instead was determined from reactor physics calculations in the same way as the Low and High libraries.

Additionally the two other libraries (Low and High) will be run with the 33 MWd/kgIHM target burnup. These runs essentially assume that the one group cross sections relevant to the Low and High burnup/enrichment conditions, respectively, can be applied to the Standard case. The results demonstrate the accuracy of the dynamically generated library compared to the Standard library and give a comparison for the value added by the dynamic library.

II.C. Forward Mode Reactor Cases

The forward mode operation test using the interpolator will be conducted by inputting a fixed fuel composition into Bright-lite using the same three libraries used in the blending mode cases. In this mode of operation Bright-lite calculates the discharge burnup and output isotopics. The fuel will be 2.5% enriched uranium and the burnup value and output compositions of each library will be recorded. Again a dynamic library will be created using the Low and High libraries.

II.D. Multi-dimensional Case

This test was done to test the accuracy of Shepard's method for use with interpolation of multiple libraries over multiple parameters.

To demonstrate that Shepard's method is capable of handling these cases when coupled with Bright-lite's reactor libraries several libraries were used to interpolate across their bounding parameter space.

Three libraries are interpolated upon and tested against a forth for testing, have the following characteristics.

1. Enrichment: 5%U235, Burnup 60 MWd/kgIHM – 14.799 n/s/flux
2. Enrichment: 7%U235, Burnup 100 MWd/kgIHM – 13.333 n/s/flux
3. Enrichment: 9%U235, Burnup 100 MWd/kgIHM – 13.773 n/s/flux
4. Test Value: Enrichment 6, Burnup 60 MWd/kgIHM – 14.650 n/s/flux

Previous tests were integral evaluations in the sense that they compared fuel blending, burnup and criticality results, all of which depend on multiple data libraries acting together. For this test the purpose is to visualize the outcomes of the multidimensional interpolation on a single library parameter as well as to compare interpolation outcomes for that parameter alone. Hence only the neutron production rate (in n/s/unit flux) the start of reactor operation from each library is interpolated upon.

III. RESULTS

III.A. Blending Mode Reactor Cases

Selected isotopics of the post-burnup composition of the blending case defined in Section II.B can be seen in Figure 5. Let the discharge mass of isotope i if the appropriate 'Standard LWR' library is used be M_i. Let the discharge mass of that isotope if another library is used (either 'Low LWR', 'High LWR,' or 'Dynamically Interpolated' as shown in the legend) be m_i. Then the percent difference, d, illustrated in Figure 5 and succeeding figures is given by

$$d = \frac{|M_i - m_i|}{M_i} \qquad 5$$

They show that the interpolation method, when applied to two libraries applicable to conditions relatively distant from the test case, produced discharge compositions that are within 10% of the most appropriate library's values. The dynamic library also showed good improvement over using either of the distant (low or high) libraries to model the targeted parameters (3% U235, 33 MWd/kgIHM).

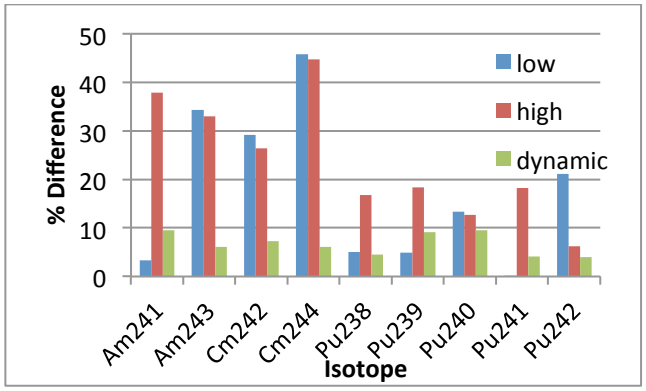

Figure 17 Results of Blending mode interpolation test

In addition to tracking the isotopic compositions of the spent fuel, the U235 enrichment of the input fuel, which is calculated by Bright-lite in order to achieve an appropriate end-of-equilibrium cycle multiplication factor, was also recorded for each case.

The results are listed in Table II. The dynamically interpolated library is seen to have found almost exactly the correct initial enrichment.

TABLE II

The U235 enrichment values for the reactors input fuel.

Library	Value	% Diff with respect to Standard library
Standard	3	0
Low	2.9	3.33
High	3.76	25.3
dynamic	2.98	0.67

III.B. Forward Mode Reactor Cases

Figure 6 shows the post-burnup composition of the forward mode case defined in Section II.C. Recall that in this case the initial enrichment is specified and Bright-lite determines the cycle length. Again the library interpolation method matched the target library to within 10% for all isotopes tested. There is once again good improvement over using just the low or high libraries to simulate the targeted reactor parameters.

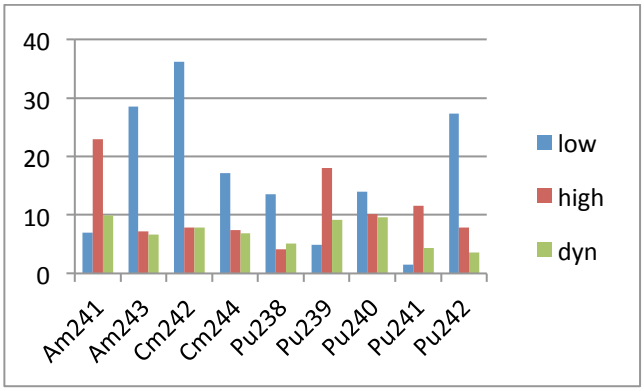

Figure 18 Results of the forward mode interpolation test

For this test the equilibrium discharge burnup calculated by Bright-lite given 3.0% enriched fuel is shown in Table III. The interpolation method performed well in this regard showing excellent agreement with the target library.

TABLE III. Burnup values given 3.0% U235 fuel.

Library	Value [MWd/kg]	% Diff with respect to standard library
Standard	33	0
Low	34.36	4.12
High	29.81	9.67
dynamic	33.11	0.33

The dynamic interpolation between libraries applicable to distant state points to create a library that is appropriate for a case of interest has been shown to lead to consistent improvements when compared against simply using the nearest available library. But the results shown here indicate that the target library is in this case probably too far from either of the two libraries used in the interpolation. A methodology for quantifying the accuracy of the interpolation method when determining how widely to space the libraries is being developed.

In both tests the difference in the accuracy between the isotopic compositions and the enrichment/burnup values comes as a result of the one group cross sections underlying each of these libraries. Because the heaviest transuranics require several capture reactions to form, any variance in their one group absorption cross sections or those of their many parents can have a big impact on their concentrations in spent fuel. The unique behavior of the cross sections of each transuranic means they do not change monotonically with a change in the reactor parameters.

III.C. Multi-dimensional Case Results

Figure 7 visualizes the results of the multi-dimensional case. The figure shows a surface of interpolated points made using three libraries (those highlighted in green). The value tested against (Enrichment 6%, Burnup 60 MWd/kgIHM) is indicated with a black box. The values in the figure are the neutron production rates (n/s/flux) of Uranium 235 in these libraries at the very first time step (i.e. zero fluence).

Burn up	Enrichment % U235								
MWd/kgIHM	5	5.5	6	6.5	7	7.5	8	8.5	9
60	14.8	14.6	14.4	14.3	14.2	14.1	14.1	14.0	14.0
62	14.7	14.6	14.4	14.3	14.2	14.1	14.0	14.0	14.0
64	14.6	14.5	14.4	14.2	14.1	14.1	14.0	14.0	14.0
66	14.5	14.5	14.3	14.2	14.1	14.1	14.0	14.0	14.0
68	14.5	14.4	14.3	14.2	14.1	14.0	14.0	14.0	13.9
70	14.4	14.3	14.2	14.1	14.1	14.0	14.0	13.9	13.9
72	14.3	14.3	14.2	14.1	14.0	14.0	13.9	13.9	13.9
74	14.3	14.2	14.1	14.1	14.0	14.0	13.9	13.9	13.9
76	14.2	14.2	14.1	14.0	14.0	13.9	13.9	13.9	13.9
78	14.2	14.1	14.0	14.0	13.9	13.9	13.9	13.9	13.9
80	14.1	14.1	14.0	13.9	13.9	13.9	13.8	13.8	13.8
82	14.1	14.0	14.0	13.9	13.9	13.8	13.8	13.8	13.8
84	14.0	14.0	13.9	13.9	13.8	13.8	13.8	13.8	13.8
86	14.0	13.9	13.9	13.8	13.8	13.8	13.8	13.8	13.8
88	13.9	13.9	13.8	13.8	13.7	13.7	13.8	13.8	13.8
90	13.9	13.9	13.8	13.7	13.7	13.7	13.7	13.8	13.8
92	13.9	13.8	13.7	13.7	13.6	13.7	13.7	13.8	13.8
94	13.9	13.8	13.7	13.6	13.6	13.6	13.7	13.7	13.8
96	13.8	13.8	13.7	13.6	13.5	13.6	13.7	13.7	13.8
98	13.8	13.8	13.7	13.5	13.4	13.6	13.7	13.7	13.8
100	13.8	13.7	13.7	13.5	13.3	13.5	13.7	13.7	13.8

Figure 19 Results of the interpolation method of on the neutron production rate of U235 at the first fluence step

The advantage of using an inverse distance system becomes apparent by looking at Figure 7. The interpolation surface is represented by the heat map. This map was created using only three libraries whose locations are shown by the highlighted boxes.

The interpolated value at the same state point (6% U235 – 60 MWd/kgIHM) as the fourth test library is approximately 14.410 n/s/flux. A comparison between this and the value of 14.650 n/s/flux in the library derived using physics calculations for this state point shows a less than 1.7% difference.

IV. CONCLUSIONS

The interpolation scheme chosen for Bright-lite's reactors provides a method of generating the libraries used by Bright-lite without resorting at runtime to external radiation transport software such as OPENMC. This provides both a computation time savings and also a memory storage savings.

The limitations to the method demonstrated in this paper are indicators that more work needs to be done to determine the accuracy range of the system to ensure that the libraries generated by this method are valid and not returning erroneous results.

This system shows good promise for providing the Bright-lite software and by extension the Cyclus fuel cycle simulator with the ability to simulate a wide range of reactors using a small set of libraries. This is important for simulating not only the widely varied current reactor fleet but also future reactors that are being investigated. Properly modelling both of these types of reactors

allows a simulator to accurately show the transition between current and future reactors, and help determine the best course of action for the future of nuclear power.

V. FUTURE WORK

A method for precalculating new Bright-lite library sets is under development. This method will augment the family of one group cross section libraries which had been converted to the fluence-dependent data needed by Bright-lite using ORIGEN 2.2.

This new methodology and software is called XSGEN. This software couples OPENMC [6] and ORIGEN2.2 to create burnup-dependent multigroup and one group cross sections and then new Bright-lite libraries.

OPENMC is a neutron transport solver that uses Monte Carlo methods like MCNP. This software generates the one group crosses sections required for ORIGIN2.2. ORIGEN2.2 performs the burnup calculations used to generate the data required for Bright-lite's libraries.

Since these cross sections are generated under the guidance of a user, the exact values of all of the parameters used in generating a library can be recorded. This will give the interpolation tool not only more accurate measure of the values of each library's parameters but also a wider scope of parameters to use.

Using XSGEN will allow for expanded testing of the interpolation method using a larger number of parameters and a different set of parameters.

This will open up the possibility of expanding this work to isotopic blending more complicated than two isotope blending. This is required for more advanced reactor types that take spent nuclear fuel TRU as input fuel or use unique fuel forms. Additionally it will allow for matching on other important measure in the nuclear fuel cycle; conversion ratio, non-leakage probability, reactor geometries, batch number, flux, and any other interpolation parameter the user records for their reactor.

The key benefit of the XSGEN tool is that it allows for the automatic generation of Bright-lite libraries. A tool to couple Bright-lite and XSGEN will be used to generate a suite of libraries for a particular reactor design. This tool will ensure that a suit is comprised of all of the necessary libraries to ensure that the Shepard's method interpolator is accurate to within 5% within the parametric space.

For example a suite of libraries can be created for a Westinghouse 17x17 reactor design with varying parameters for burnup [20-60MWd/kgIHM], enrichment[2, 5% U235]. The automatic library generation tool will create the necessary libraries to ensure that the interpolation method works over the entirety of the interpolation space.

REFERENCES

[1] R. Flanagan, C. Bagdatlioglu and E. Schneider, "Fuel Composition Techniques of Nuclear Fuel Cycle Simulators," in *Physor 2014 - The Role of Reactor Physics Toward a Sustainable Future*, Kyoto, Japan, 2014.

[2] Cyclus, "Cyclus Fuel Cycle Simulation," 2013. [Online]. Available: www.fuelcycle.org. [Accessed Feburary 2014].

[3] O. R. N. Laboratory, *Origen 2, Version 2.2,* Oak Ridge, Tennessee, June 2002.

[4] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *ACM Nat. Conf.*, 1968.

[5] W. Gorgon and J. Wixom, "Shepard's Method of "Metric Interpolation" to Bivariate and Multivariate Interpolation," *Mathematics of Computation Volume 32, Number 141,* pp. 253-264, 1978.

[6] MIT, "OpenMC User's Guide," MIT, 2014. [Online]. Available: http://mit-crpg.github.io/openmc/usersguide/index.html. [Accessed August 2014].

Appendix D

Citation: C. Bagdatlioglu, R. Flanagan, E. Schneider, "Characterizing the United States Nuclear Used Fuel Using Medium Fidelity Reactor Modeling Software," ICONE 24, Charlotte, NC, June 26-30 2016 (Accepted)

CHARACTERIZING THE UNITED STATES NUCLEAR USED FUEL INVENTORY USING THE CYCLUS FUEL CYCLE SIMULATOR

Cem Bagdatlioglu
University of Texas at Austin
Austin, Texas, USA

Robert Flanagan
University of Texas at Austin
Austin, Texas, USA

Erich Schneider
University of Texas at Austin
Austin, Texas, USA

ABSTRACT

The used fuel inventory of the United States commercial nuclear fleet has been accumulating since the inception of nuclear reactors. In order to understand the mass and composition of the used fuel inventory, a nuclear fuel cycle simulation package (Cyclus) is used with a reactor modeling tool (Bright-lite). The parameters for the simulation are obtained as historical operation and burnup data for every reactor in the US fleet, taken from the U.S. Energy Information Administration. The historical burnup data is used to calculate the fuel enrichment of every reactor at every refueling. Discharged uranium inventories calculated by the software are shown to closely match the reference data. The total mass of three major actinide groups are presented as they build up over time. In addition, the evolution of the plutonium composition in discharged fuel is also presented, illustrating Cyclus' ability to track the composition of material flowing through a large, evolving reactor fleet over decades.

INTRODUCTION

Since the advent of nuclear energy, used fuel from commercial reactors in the United States has been accumulating. Understanding the composition of this used fuel is important for reasons ranging from waste management to material accountability. Simulation modeling of the historical U.S. reactor fleet can be employed to calculate the mass of each isotope currently in used nuclear fuel.

The work presented here employs a nuclear fuel cycle (NFC) simulation package, Cyclus [1], with a medium fidelity reactor modeling tool, Bright-lite, to simulate the historic U.S. nuclear reactor fleet from beginning of operation until present day. Its objective is to demonstrate the capabilities of Cyclus to arrive at a reasonable representation of the used fuel inventory of a large, complex reactor fleet.

Bright-lite, a nuclear reactor fuel blending and burnup simulator, was developed to be a module of Cyclus [2]. The methodology used within Bright-lite is capable of determining the output fuel composition and burnup of a nuclear reactor for a given fuel loading, as well as calculating a blending ratio for input fuel streams given constraints such as burnup, or conversion ratio given that criticality must be maintained through the cycle. In order to calculate the input composition of a fuel batch for which one or more of these constraints are specified, Bright-lite takes two fuel streams and finds the blending ratio for the streams so that the resulting fuel will meet all constraints. For light water reactors, streams of U-235 and U-238 are blended to match a target burnup, simulating the process of selecting an enrichment to obtain a desired cycle length and burnup at a given power density.

To enable rapid execution, Bright-lite uses pre-calculated libraries that tabulate the transmutation, burnup, and neutron production and destruction rates of every isotope that may be present in initial fuel loadings. These libraries are then used dynamically during runtime to simulate arbitrary combinations of those isotopes.

Bright-lite is a strong tool for reactor fleet characterization because it can model the transients of reactor startup and shutdown as well as quasi-equilibrium refueling with slowly time-varying refueling batch enrichments and burnups. This allows

the system to model the evolution of the U.S. reactor fleet from start to finish including improvements in availability and the increase in average discharge burnup over time.

METHODOLOGY

The task of modeling each reactor within the U.S. nuclear fleet requires historical operation and burnup data for every reactor. This data was accessed through the U.S. Energy Information Administration (EIA) [3][4][5]. For each reactor the following information was used.

- Startup date
- Scheduled shut down date
- Lifetime extension data
- Time dependent burnup
- Time dependent availability
- Reactor thermal power
- Reactor core mass

In this source only averaged values of time dependent burnup and availability are provided for reactors in the U.S. fleet. For instance, burnup data for two broad reactor types (PWR and BWR) is presented on an annually-averaged basis for all fuel discharged in each year. The general trend of this data is an increasing burnup every year until the last available data point. Since the data shows some year-to-year volatility, the burnups were averaged over 5-year intervals to provide the data series used by Bright-lite.

Bright-lite utilizes this data by determining a unique target burnup at every refueling and calculating the necessary fuel enrichment to achieve the target. The trend of increasing burnup therefore translates into a parallel trend of increasing enrichments and longer cycle lengths. In the model, the target burnup during refueling is calculated by interpolating on the two nearest-neighbor 5-year average burnup points.

The fuel enrichment calculated by Bright-lite will yield the desired burnup at the equilibrium cycle. When the new fuel is loaded in reactor cores, the transient burnup calculation also accounts for other batches in the core which were loaded in earlier cycles. These previous batches may have lower enrichments due to lower target burnups at the time of their loading, causing the reactor discharge burnup to meet the target after several cycles.

The burnup curves for each reactor type can be seen in Figure 1.

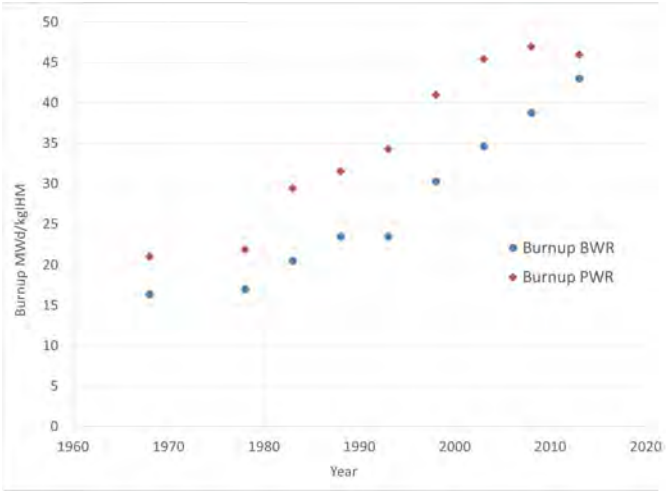

Figure 20 – Historical burnups for the US nuclear fleet.

The burnup data for the reactors was modified for the period 1960-1980. This modification was driven by the inclusion of partially-burned startup batches in the EIA data when determining the average burnup for a given year. To account for the startup batches during this period the number of startup batches was counted and used to elevate the burnups during these years using the following equation.

$$x = \frac{(S_1 * 0.33 + S_2 * 0.66 + S_3)}{S_1 + S_2 + S_3} \qquad (1)$$

Here three-batch refueling is assumed and S_1 is the number of first time outages, S_2 is the number of second time outages, and $S_3$is the number of normal outages during the time period. This will account for the fact that usually the first and second batch out of a reactor will have approximately 33% and 66% of the burnup that a steady state reactor might have.

This correction will return a fractional value that can be used to scale up the burnups using the following equation.

$$BU_{new} = \frac{BU_{old}}{x} \qquad (2)$$

BU_{new} will reflect the targeted steady state burnup of each reactor type during a specific time period, adjusting upward the actual averaged burnup over that time period from the EIA data.

For PWRs, three Bright-lite libraries associated with different discharge burnups (low: 20 MWd/kgIHM, standard: 33 MWd/kgIHM, high: 50 MWd/kgIHM) were used to model the burnup as it evolved through time. The built in interpolation tool [6] handled the interpolation and generated a new library for the reactors each time the reactor refueled with a different burnup.

The same 5 year averaging process was applied to the EIA data for the average availability of the BWR and PWR reactor types. This curve can be seen in Figure 2. Bright-lite uses this availability data to adjust the outage time of reactors: instead of introducing random mid-cycle outages, Bright-lite simply adjusts the refueling outage times such that the trend in long-run average availability is correct.

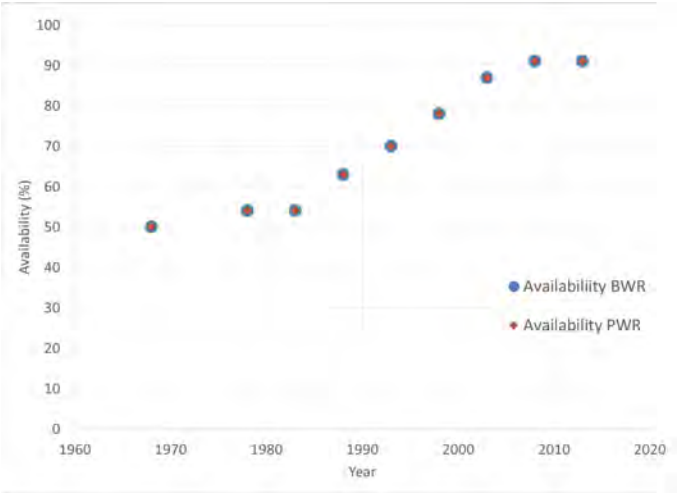

Figure 21 – Historical availability of the US nuclear fleet.

The heavy metal mass and average power density of each reactor were not directly available from the EIA data. The core masses were therefore determined by obtaining an average mass per fuel assembly for each reactor type. This was obtained from the mass of heavy metal and the number of assemblies being refueled annually to each reactor type. The total mass was then divided by the number of assemblies for each type to obtain a mass per assembly. These masses were next multiplied by the total number of fuel assemblies in each reactor to determine the heavy metal inventory of the reactor. This process treats every reactor of a given type as having the same heavy metal mass per assembly, so the results presented here are not expected to be accurate for each individual reactor. On the other hand, the cumulative results summed over all reactors should be fair representation of the fleet.

RESULTS

Figure 3 shows the steady state enrichments calculated by Bright-lite based on the target burnup for that year. The increasing trend in these enrichments follows the increase in reactor burnups.

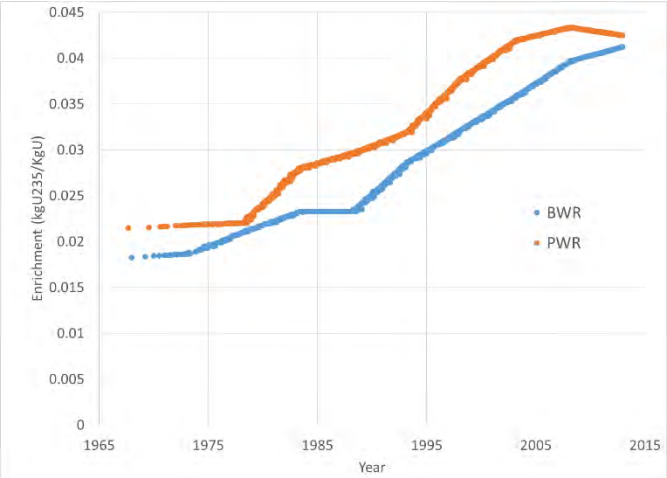

Figure 22 – Batch enrichments calculated by Bright-lite.

Figure 4 shows the discharge burnup of every batch in the Bright-lite simulation. The historical US fleet data (marked 'EIA') is overlaid on this graph for comparison. Note that the Bright-lite data shows the burnups of all batches, including batches associated with reactor startups and shutdowns.

It can be observed from this figure that during time periods where the historical burnup change is low, the Bright-lite burnups begin to match the target very closely. During periods where the target burnup changes relatively quickly the Bright-lite discharge burnup either lags or leads, as actual cycle burnups are calculated based upon the varied initial enrichments of all batches which are in-core.

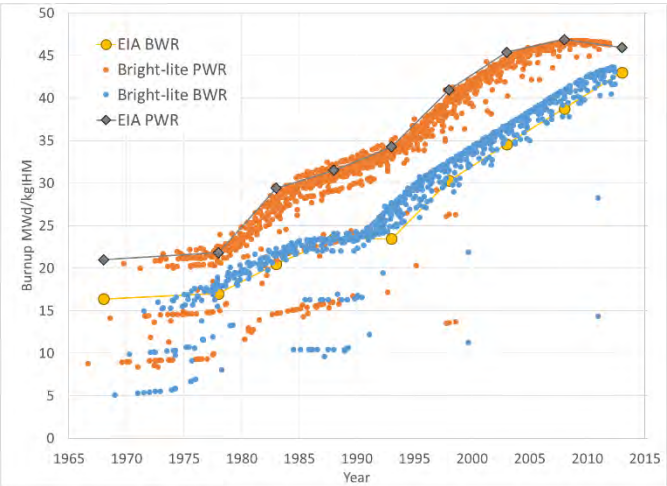

Figure 23 – Discharge burnups from Bright-lite simulation.

Bright-lite cycle burnup calculations have been compared against several reference data sets [8], but these validation exercises have focused fuels with on modern burnup levels. It is thus useful to also compare the results shown here against older data. According to a 1980s report by Oak Ridge National Lab [7], the approximate enrichment for a BWR with burnup of 27 MWd/kgIHM was 2.75 %. Figure 3 and 4 show good agreement with this for the BWRs. Additionally, the report shows that for a PWR with burnup of 33 MWd/kgIHM the steady state enrichment should be 3.2 %. Again, Bright-lite demonstrates good matching for this case.

Figures 5 and 6 show the total mass of initial heavy metal discharged from each reactor type as a function of time. It can be seen that the EIA trend is followed for each reactor type. Minor differences in these figures are in part caused by the method for calculating a reactor's core mass. Also, no attempt was made to match specific refueling outage dates for individual reactors.

Therefore, the same reactors may not be refueling in the same years between Bright-lite and the EIA data. While this leads to errors in individual years, the cumulative amount of fuel discharged remains quite accurate.

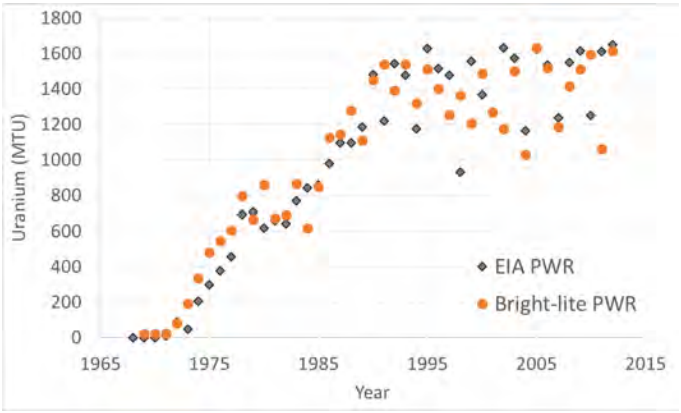

Figure 24 – Total uranium discharge per year, PWR's.

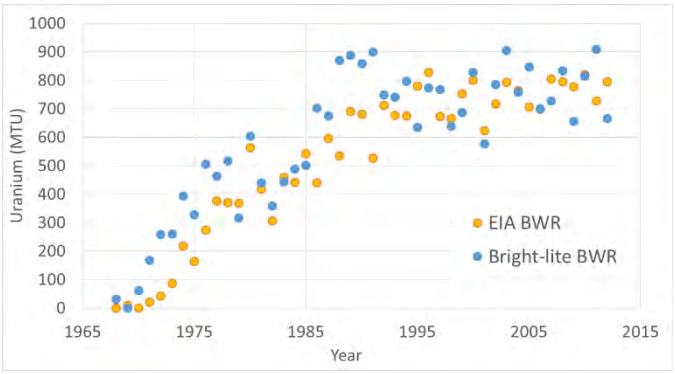

Figure 25 – Total uranium discharge per year, BWR's.

Figure 7 shows the total mass of three actinide elements (Pu, Am, and Cm) contained in discharged fuel as a function of time. These curves take into account radioactive decay. The PWRs show higher total mass for each actinide because the total mass of fuel going into the PWRs each year is higher than for the BWRs (with the exception of the first fuel years).

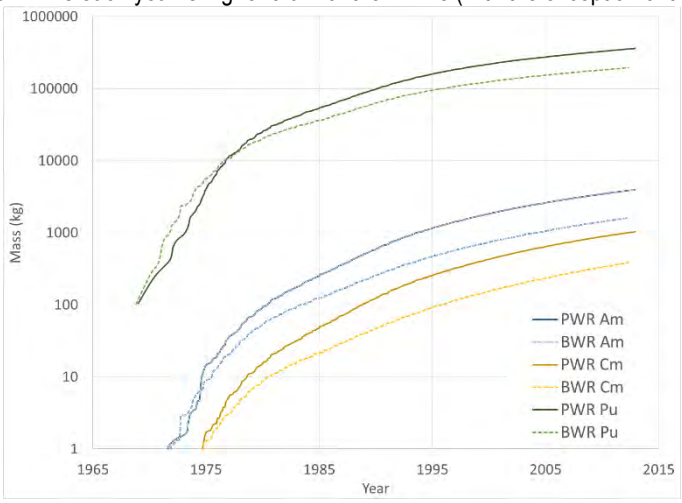

Figure 26 – Total mass of Am, Cm, and Pu.

Figure 8 shows the discharged plutonium and fissile Pu over the total mass of uranium entering the core for each discharged batch for both the BWRs and PWRs. The behavior seen here is to be expected as the BU of the fuel increases. Early deviations from the trend are caused by the startup transient batches and later outliers arise from reactor shutdowns.

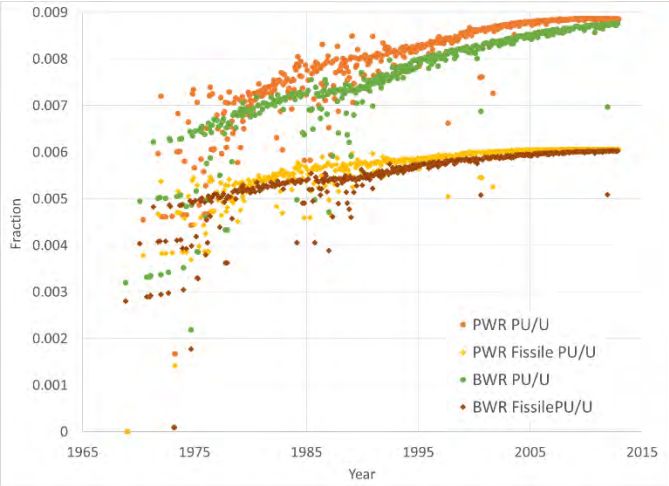

Figure 27 – Pu and fissile Pu fractions over time.

CONCLUSION

Bright-lite was used to simulate the US nuclear reactor fleet from the first reactors (early 1960s) until 2012. Discharge burnup data, grouped for BWRs and PWRs, was used to determine a target burnup for each reactor at every refueling. The target burnups for a given reactor was calculated by interpolating on the two nearest-neighbor 5-year average burnup points.

Burnup trends and discharge material masses were shown to closely match the reference data. This qualitative comparison indicates good matching between Bright-lite and the actual data.

Total mass of three major actinide groups, as well as plutonium discharge information were presented. These results showed expected behavior consistent with the reactor deployment schedules.

Future work will seek to validate these results against those generated by other simulations of the historical US reactor fleet.

REFERENCES

[1] CNERG Fuel Cycle Group, "Cyclus," 01 01 2014. [Online]. Available: http://fuelcycle.org/. Web. 01 Jan. 2014.

[2] "Fuel Composition Calculation Techniques of Nuclear Fuel Cycle Simulators," PHYSOR 2014, The Westin Miyako, Kyoto, Japan, 28 Sep. – 03 Oct. 2014, on CD-ROM (2014)

[3] "Table 3. Nuclear Reactor Characteristics and Operational History" *Nuclear & Uranium*. U.S. Energy Information Administration, 11 Nov. 2011. Web. 14 Dec. 2015.

[4] "Table 1. Nuclear Reactor, State, Type, Net Capacity, Generation and Capacity Factor" *Nuclear & Uranium*. U.S. Energy Information Administration, 11 Nov. 2011. Web. 14 Dec. 2015.

[5] "Table 3. Annual Commercial Spent Fuel Discharges and Burnup, 1968- June 30, 2013" *Spent Nuclear Fuel*. U.S. Energy Information Administration, Dec. 7 2015. Web. 14 Dec. 2015.

[6] "Multidimensional Cross Section Library Interpolation for the Bright-lite Reactor Modeling Software," Robert Flanagan, E. Schneider, C. Bagdatlioglu; Global, 2015.

[7] "Standard- and Extended-Burnup PWR and BWR reactor models for the ORIGEN2 Computer Code" S.B. Ludwig, J.P. Renier, Oak Ridge National Laboratory, Jan 3, 1990.

[8] "Using Spatial Flux Calculations to Improve the Fluence-Based Neutron Balance Approach," Cem Bagdatlioglu, Erich Schneider, Robert Flanagan; ANS Annual 2015.

Appendix E

Citation: C. Bagdatlioglu, R. Flanagan, E. Schneider, "Fuel Cycle Analysis Using Bright-lite in the Cyclus Simulator," Physics of Rectors Conference (PHYSOR) 2016, Sun Valley, Idaho, May 1-5. (Accepted)

FUEL CYCLE ANALYSIS USING BRIGHT-LITE IN THE CYCLUS SIMULATOR

Cem Bagdatlioglu, Robert R. Flanagan, Erich E. Schneider

Department of Nuclear & Radiological Engineering

University of Texas Austin, Texas, USA

cemb@utexas.edu

ABSTRACT

Inherent to fast reactor fuel cycles is a variance in the amount and composition of transuranic material available to fast reactors as input fuel. This variance requires Bright-lite to dynamically update blending fractions of incoming fuel batches to ensure that reactor constraints are met. As part of this process Bright-lite also determines the output compositions unique to each input fuel batch. This work demonstrates the capabilities of the Bright-lite reactor modeling methodology that allows it to simulate fast reactor closed fuel cycles as they evolve through a 200 year time frame.

Key Words: **fuel cycle, Cyclus, reactor modeling.**

1. INTRODUCTION

Accurate tracking of materials as they move through reactors is necessary to evaluate nuclear fuel cycles (NFC) and reactor technologies as well as the metrics derived from fuel cycle material balances. NFC simulators enable the study of fuel cycle and reactor technology options by informing users on metrics such as future supply and demand of materials, energy production, and waste inventories. Calculation of reactor material balances in a NFC simulator can become very complex when the cycle involves many reactor technologies incorporating fuel recycling.

One of these NFC simulators is Cyclus [1], which allows independent developers to plug-in their software to work within its architecture. Using Cyclus, developers can add models which enhance fidelity or detail relating to specific components in the NFC (such as improving reactor physics or optimizing waste inventories) without the burden of coding a complete simulator.

2. OVERVIEW OF BRIGHT-LITE

A reactor module, Bright-lite, provides Cyclus with a fast, medium fidelity capability for reactor material balance calculations. Bright-lite determines fuel isotopics and burnup of nuclear reactors using a neutron balance approach to obtain viable fresh fuel compositions when multiple feed streams are present. This method uses pre-calculated fluence-based libraries and the initial fuel compositions to evaluate fuel depletion. It enables Bright-lite to execute quickly while maintaining accuracy and applicability across various reactor technologies.

Bright-lite reactors can be operated in two modes, presented in Figure 1. In 'forward' mode reactors take a given fuel composition and determine the output isotopics and burnup of the fuel. In 'blending' mode the reactor tries to achieve a given target burnup by finding the blending ratios of available fuel streams (Stream A and Stream B in Figure 1).

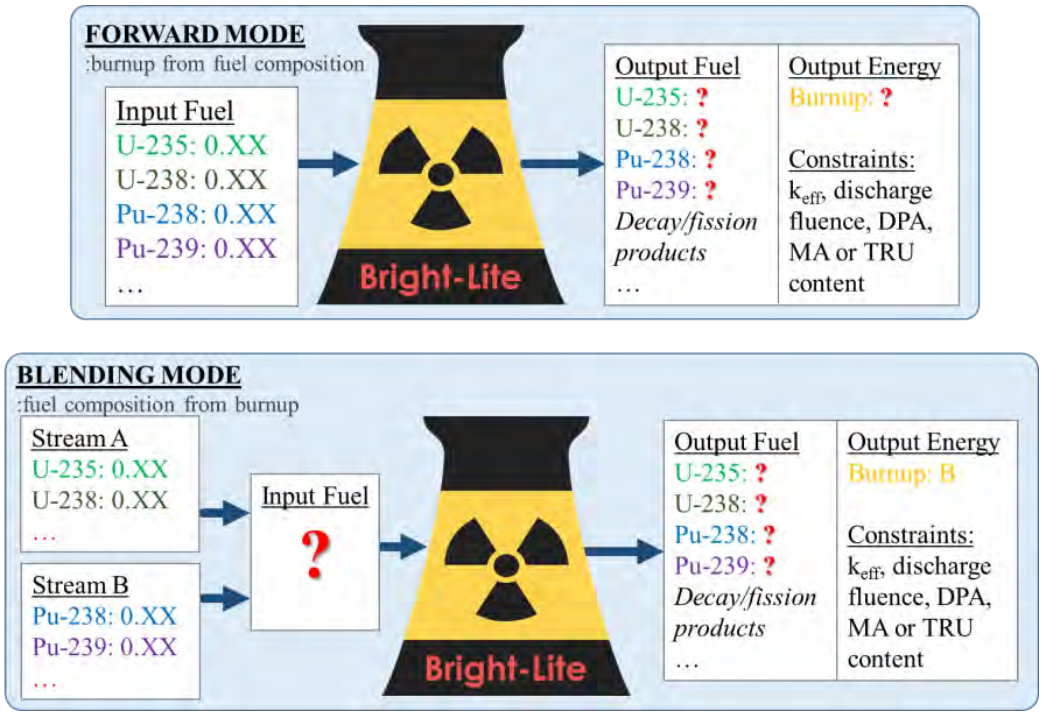

Figure 28. Two modes of operation for Bright-lite.

Bright-lite relies on pre-calculated one group cross section libraries and fuel composition vectors for burnup and transmutation calculations. The fuel composition is determined by the NFC

simulator if Bright-lite is operating in forward mode. In blending mode, however, an iterative approach is used to modify the blending stream ratio until a constraint (e.g., on multiplication factor at a given burnup or cycle length) is met.

Figure 8 depicts an example 3 batch reactor operating in steady-state. The fluence of the batches are marked on the y-axis, where $F^{(1)}$ is the fresh batch and $F^{(3)}$ has been in the reactor for two cycles. A time step Δt is selected to move fluence forward. At every time step the flux of each batch is determined (based on fluence-dependent material properties) and fluence of batches are increased. Two methods are available for calculation of batch fluxes. The simpler method assumes that each batch has a constant and equal power density. This assumption has shown to produce reasonable results for several single-fueled reactor types [2]. Alternatively, a finite difference diffusion equation solver which is integrated into Bright-lite can be used to determine spatially-averaged fluxes across any number of fuel regions. This approach allows Bright-lite to treat reactors where macro regions can operate at very different power densities, such as driver-blanket breeder or converter reactors.

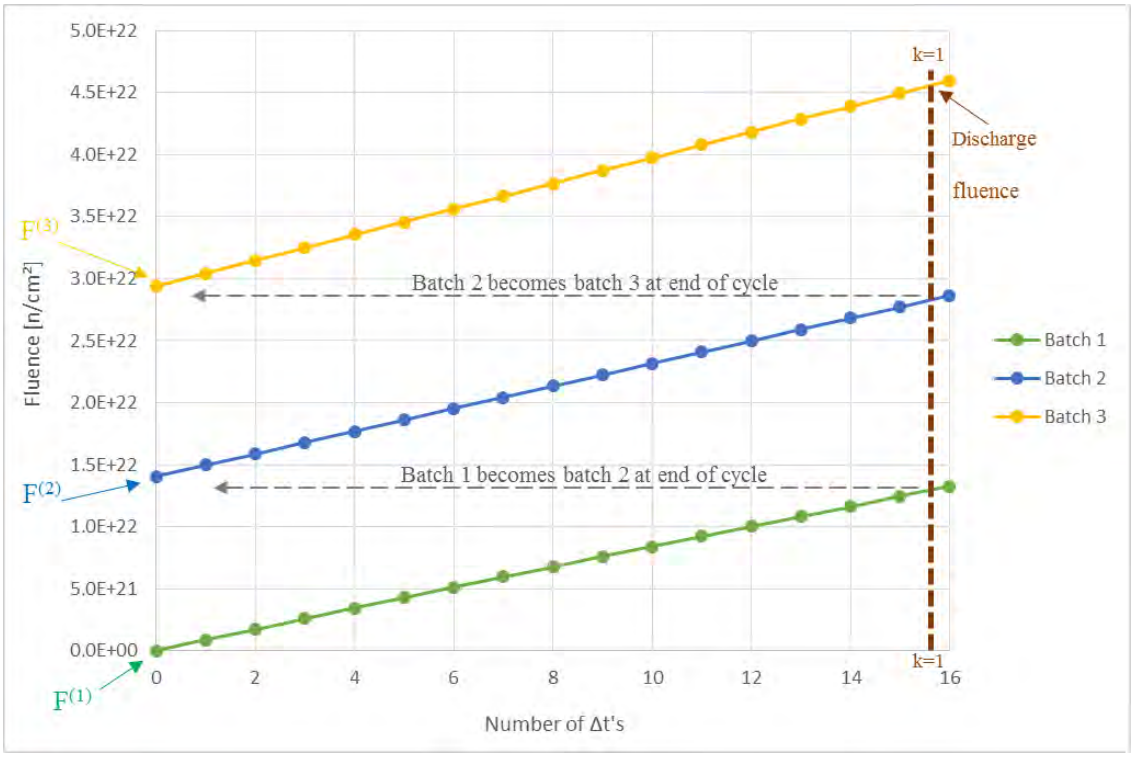

Figure 29. Discharge fluence determination for an example 3-batch LWR reactor.

Next, the criticality of the core is calculated using a calculated or user input nonleakage probability (P_{NL}) of the reactor, the fluence and batch-dependent neutron production rate ($P^{(n)}$)and neutron destruction rate ($D^{(n)}$) per unit flux of the actinides and their daughters, and flux. The effects of structural materials and non-actinides upon the reactor neutron balance are also accounted using a reactor-specific structural material composition and relevant cross sections. These effects modify the neutron production and destruction rates and are fluence-independent.

$$k(t+\Delta t) = P_{NL} \cdot \frac{\sum_{n=1}^{N} P^{(n)}(t+\Delta t) \cdot \phi^{(n)}(t+\Delta t)}{\sum_{n=1}^{N} D^{(n)}(t+\Delta t) \cdot \phi^{(n)}(t+\Delta t)} \quad (46)$$

The simulation marches through fluence until the core criticality drops below unity (or until the fluence accumulated by the oldest batch reaches a user-defined target). Once the target has been reached the end-of-cycle fluence of each batch is determined, the oldest batch is discharged, and the isotopic composition of that batch is returned to the Cyclus simulator.

Blending mode incorporates this methodology to determine the necessary input fuel blending ratio to achieve a fluence or burnup target. Figure 5 depicts this process. For example, a MOX reactor may have streams of Pu and DU available for blending. The Pu+DU blending ratio will be guessed and used to operate Bright-lite in forward mode, iterating on the blending ratio until the targeted burnup is achieved in a steady-state reload operating mode. The process repeats until a composition yielding a burnup within a threshold of the target is found.

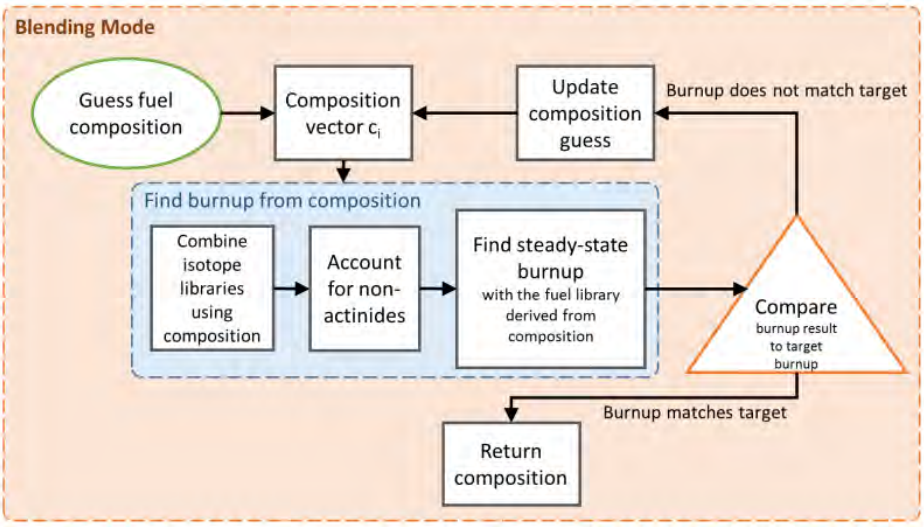

Figure 30. Input composition finding algorithm of Bright-lite.

3. FUEL CYCLE STUDY USING BRIGHT-LITE

Bright-lite's capability to simulate different types of reactors is illustrated by a simple fuel cycle transition scenario. This cycle is similar to a transition to EG24 from the fuel cycle options study [3] which involves continuous recycle of TRU-based fast reactor fuel. In the illustrative scenario a fleet of LWRs is used to start up fast reactors which burn the TRU from the LWR spent fuel, as well as their own self-recycled TRU.

The specifications for the two reactor types can be seen in **Table 4**. The conversion ratio is a derived quantity for the LWR but an input for the FR. The fast reactors aim to hit a target burnup (150 MWd/kgIHM) at the conversion ratio listed in Table 4. Implicate in this constraint is that the core must maintain criticality ($k > 1$).

Table 4. Reactor specifications for the Bright-lite fuel cycle test scenario.

	LWR	**FR [4]**
Discharge Burnup (MWd/kgIHM)	42	150
Conversion Ratio	0.58	0.70
Core Mass (kg)	127,000	60,000
Batches	3	6
Core Thermal Power (MWt)	4,000	2,800
Electrical Power (MWe)	1,320	1,204
Efficiency	0.33	0.43
Lifetime (years)	40	40

The aim of this cycle is to start with approximately 6500 MWe generation capacity and increase that to 7000 MWe over the course of the simulation. These capacity numbers were chosen to be small to simulate a new fleet starting up.

The power split of the fuel cycle is seen in
Figure 31. Four to five reactors are constructed at the beginning of each of five 40-year periods within the simulation. In the first interval, only LWRs are constructed; by the last interval, only one LWR is built with the remainder of the fleet being comprised of FRs. The number of reactors chosen at each start up period was determined based on the amount of TRU available to the system. As more TRU is made available more FRs are started up. The end goal is for this system to burn out the used TRU in the system by year 200.

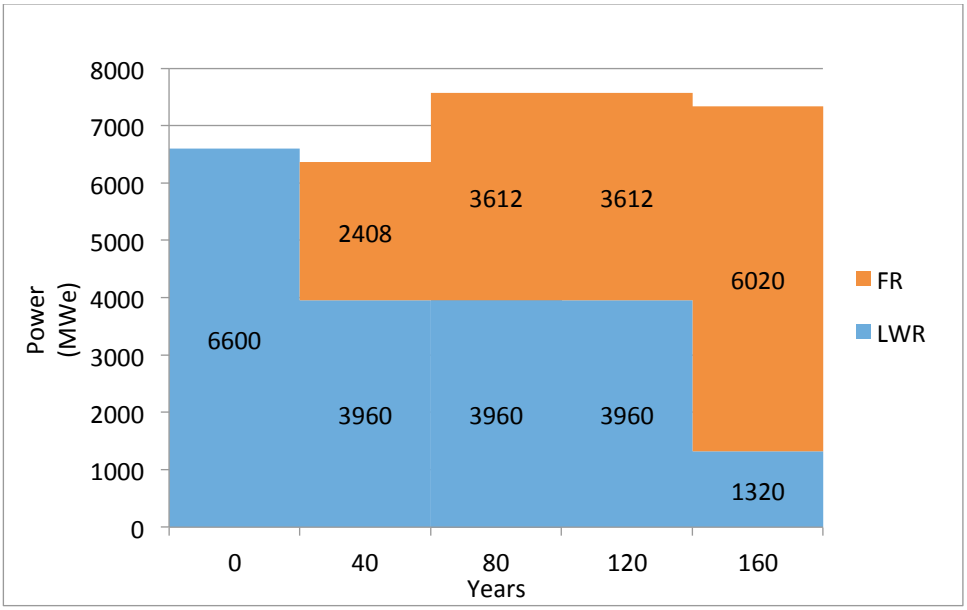

Figure 31. Power split of the reactor types in the fuel cycle.

The fast reactor plutonium charge and discharge isotopic compositions for each batch can be seen in **Figure 32**. Reading vertically, each set of five symbols represents the isotopic composition of one fuel batch. The plutonium vector evolves as the fast reactors transition from using TRU from only light water reactors to a blend which includes TRU recycled from their own used fuel.

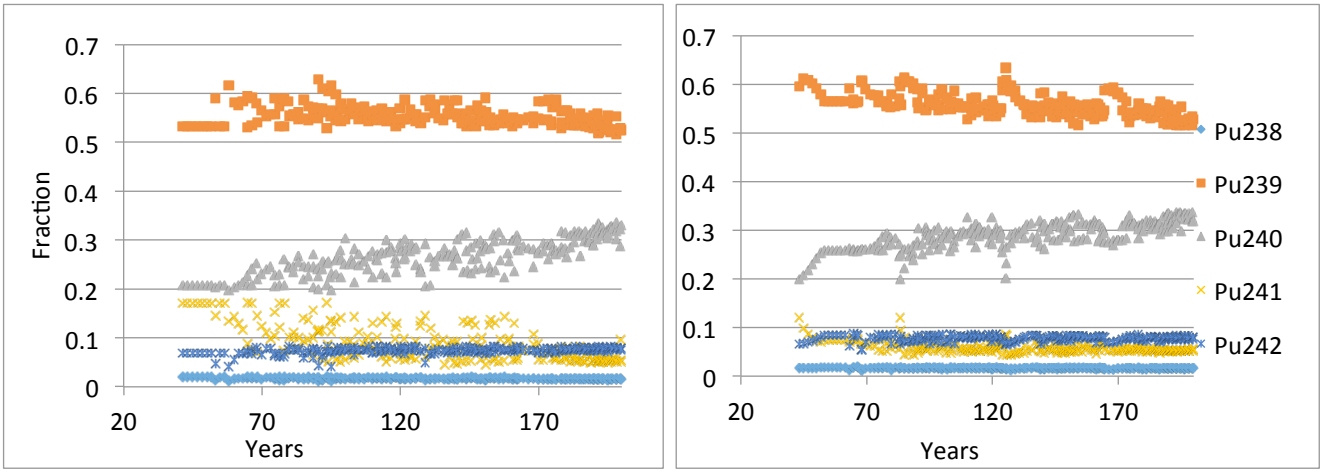

Figure 32. The concentration of plutonium isotopes in the fresh and spent fast reactors fuel.

Figure 33 shows the amount of total amount of Pu located in the system over the simulation timeline, and the amount of Pu located in the system minus the amount of Pu located in the FRs (free Pu). As more fast reactors are in operation the amount of free Pu compared to total Pu drops because Pu is being trapped in the FRs. **Error! Reference source not found.**As the fraction of fast reactors in the system increases, the slope of the growth in the free Pu decreases until it finally goes negative when the last set of reactors comes online in year 161. The large changes in Pu inventory come as a result of fast reactors starting up (drops) or LWRs or FRs shutting down (rises).

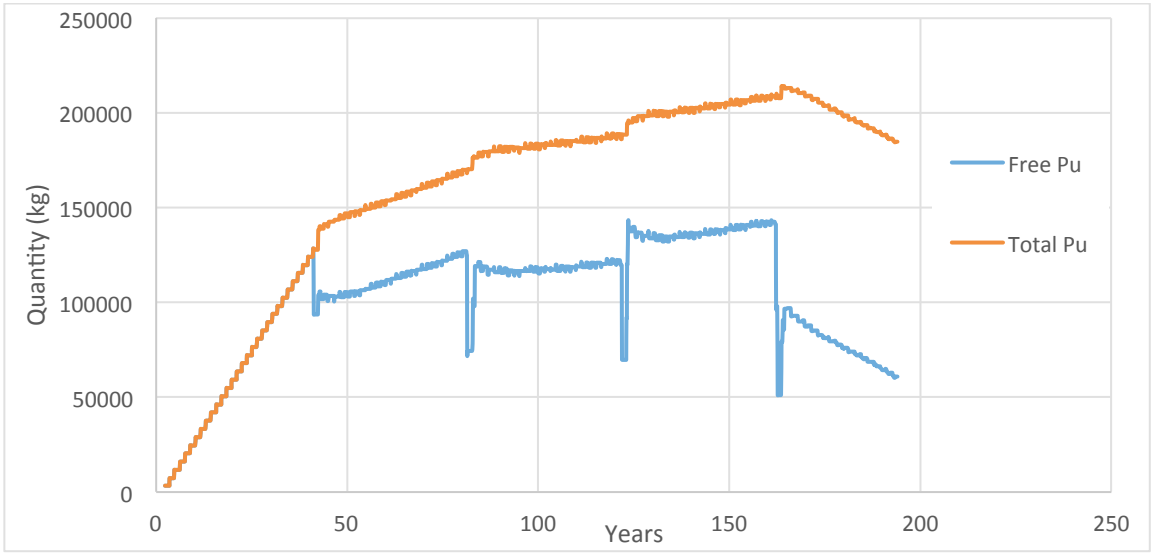

Figure 33. Free Pu and Total Pu for the duration of the simulation.

The total used TRU of each reactor type can be seen in **Figure 34**. This division shows that a majority of the used TRU in storage is LWR used TRU. This is because the system is more likely to draw from FR used TRU before it draws from LWR used TRU. Therefore, TRU from the FRs never builds up. The major cause for the drops in the LWR graph comes from the startup of new FR, because there is never enough FR used fuel in storage to start up a new FR. The fast decline in the LWR graph comes as a result of only one LWR being in operation during that period. It cannot keep up with TRU demand from the much large FR fleet.

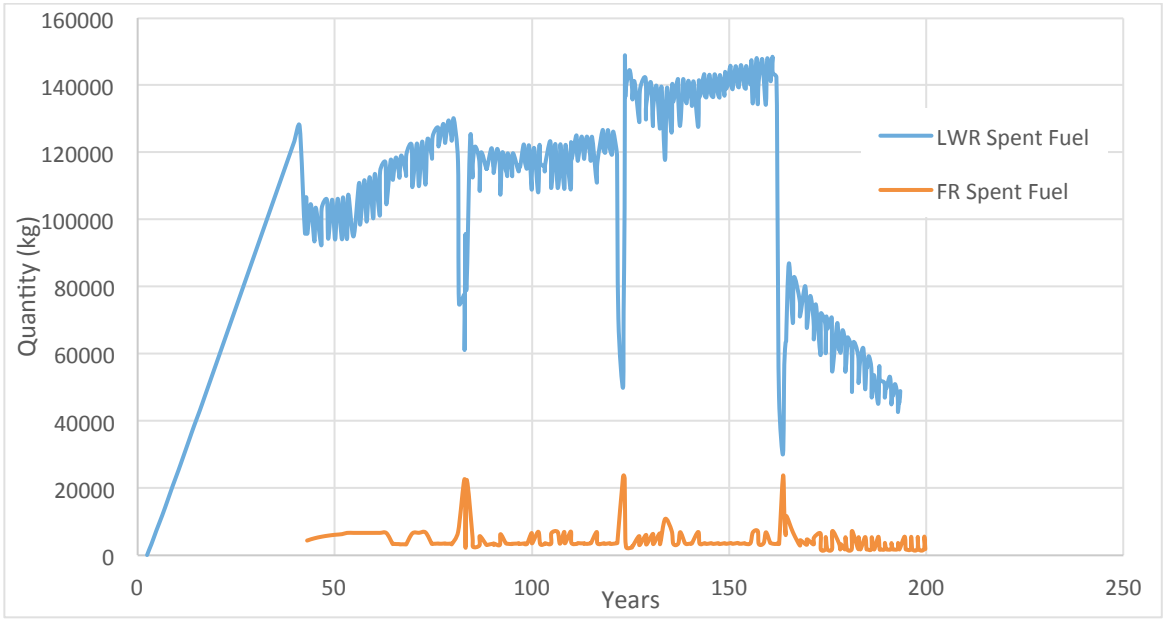

Figure 34. Spent fuel of each type located in reprocessing, fuel fabrication, or temporary storage for the test case.

4. CONCLUSIONS

The studied fuel cycle demonstrates Bright-lite's ability to provide medium fidelity modelling of reactor operation within a fuel cycle simulator. Bright-lite's ability to dynamically determine input fuel compositions from available feed streams which arise during the simulation allows it to accurately model transition scenarios. Specifically, in this scenario it demonstrates the ability to handle multiple fuel streams and still produce fuel consistent enough for steady state operation. This makes the software a useful tool for handling cycles where the composition and type of available fuel changes considerably over the course of the simulation.

REFERENCES

[1] CNERG Fuel Cycle Group, "Cyclus," 01 01 2014. [Online]. Available: http://fuelcycle.org/. [Accessed 01 01 2014].

[2] C. Bagdatlioglu, "Fluence Based Neutron Balance Approach Using Spatial Flux Calculations," (Masters thesis) 2015.

[3] U.S. Department of Energy, "Nuclear Fuel Cycle Evaluation and Screening - Final Report," 2014.

[4] Y. Ishiwatari, Y. Oka and S. Koshizuka, "Breeding Ratio Analysis of a Fast Reactor Cooled by Supercritical Light Water," Journal of Nuclear Science and Technology, 2012.

[5] M. Bell, "ORIGEN: the ORNL isotope generation and depletion code," Oak Ridge National Lab, Tenn. (USA), 1973.

[6] B. Lamarsh, Introductin to Nuclear Engineering, New Jersey: Prentice-Hall, 2001.

Appendix F

Welcome New Bright-lite User!

Bright-lite is a collection of modules for the Cyclus Fuel Cycle Simulator that allow for medium fidelity reactor modeling. There are currently three modules in the Bright-lite suite.

- Bright-lite Reactor Facility
- Bright-lite Fuel Fabrication Facility
- Bright-lite Reprocessing Facility

Bright-lite Reactor Facility: A reactor modeling software that uses burnup, criticality, and transmutations matrix curves to determine input and output isotopic compositions. The reactor can also operate with user-defined fuel compositions, called the "forward" mode.

Bright-lite Fuel Fabrication Facility: A module that communicates directly with the Bright-lite Reactor Facility that allows for the reactor facility to access blending functions that will determine the isotopic composition of fuel that the reactor will be using in order to match a specific constraint.

Bright-lite Reprocessing Facility: A module that seperates user-defined isotopes out of a material that is sent to it. This currently requires an external text file to indicate which isotopes should be seperated with what efficiency into which streams. The structure of this text file can be seen here.

Bright-lite works in conjuction with the Cyclus Fuel Cycle Simulator.

Bright-lite is currently only being actively supported for Ubuntu.

Installation

To use Bright-lite first you need to install it. Currently Bright-lite has the following dependencies.

- Cyclus Fuel Cycle Simulator

The following dependencies will be required in the future

- Eigen

To install Bright-lite please follow these instructions.

1. Clone the Bright-lite repository from github.

   git clone **repository url**

2. Change directory into the Bright-lite directory using the following command.

   cd Bright-lite

3. Use the following command inside the Bright-lite directory.

   python install.py

This will add the Bright-lite module to the cyclus environment, and allow you to use Bright-lite in Cyclus simulations.

Using Bright-lite

Bright-lite reactor requires at least 6 inputs from the users to operate fully. While there are several other inputs associated with the module, all of these other inputs come with a default value.

The six required inputs are

- **in_commods**: This field is a one or more that indicates the possible sources of fuel for the reactor. The values in this field should be commodities that exist inside of the simulation. In order to use Bright-lite in forward mode, set the first source to the steady-state (non-startup) fuel supplier. For forward mode the startup fuels can be set with additional sources.

- **out_commod**: This field should be filled out with the cyclus commodity that will connect the reactor facility to the facility that will be directly handling the waste.
- **libraries**: This is a one or more field that indicates the Bright-lite library database the reactor will be using. Note: Adding additionally library databases to this list will enable the library interpolation capabilities in Bright-lite but also requires that the user input parameters and values to be interpolated upon. The interpolation feature is intended for advanced users.
- **target_burnup**: This field indicates the target burnup for the reactor. If this is set to 0, the reactor will operate in forward mode. If this value is not set to zero the Bright-lite reactor must be connected to a Bright-lite fuel fabrication facility.
- **core_mass**: This field indicates the total mass of fuel inside of the core. This mass does not include structural components, it is only the mass of fuel to be burned.
- **generated_power**: This indicates the total thermal generating power of the core. The electrical generated_power will be this value times the effiency of the reactor (a input set to default at 33% but is user adjustable).

Operational Modes

Bright-lite has two operational modes. The mode is indicated using inputs to the Bright-lite reactor module. Forward mode is chosen by setting the reactor *target_burnup* mode to be equal to 0. Blending mode requires the *target_burnup* field to be a non negative value. Additionally, blending mode requires the Bright-lite ReactorFacility to be connected to a Bright-lite FuelfabFacility.

Forward Mode

In forward mode Bright-lite accepts a fuel composition and burns it. It does this by advancing the fluence of each batch in the core until the target is met (such as k = 1).

Currently forward mode works only with *criticality* and *burnup* targets.

Blending Mode

As stated above using the blending function in Bright-lite requires connecting a Bright-lite ReactorFacility to a Bright-lite FuelfabFacility. The *in_commods* field of the Bright-lite reactor should include all of the fuel fabrication facilities that the reactor can be connected to.

Currently there are two blending modes available in Bright-lite. These modes are described by a target-constraint pairing. The two available pairs currently are:

1. Burnup - Criticality: The blender will create a fuel that meets a target burnup when criticality is equal to the given constraint. This set of constraints only requires a non negative number to be entered into the *target_burnup* field.

2) Burnup - Conversion Ratio: The blender will create a fuel that meets a target burnup when conversion ratio is equal to the given constraint. This is achieved by setting the *CR_target* input field of the reactor to be equal to a number greater than 0 (note that there is no upper bound limit in the code for this this but physically it should not exceed 2). Additionally the*target_burnup* field must be a non negative value for this to work.

Running Example Cases

There are several example cases provided with Bright-lite. The single reactor example cases are:

- LWR, 3 batch, 33 MWd/kg burnup, 3.1 % U-235 fuel (forward mode)
- LWR, 4 batch, 42 MWd/kg burnup, 3.6 % U-235 fuel (forward mode)
- LWR, 4 batch, 51 MWd/kg burnup, 4.3 % U-235 fuel (forward mode)
- LWR, 4 batch, 45 MWd/kg target burnup (blending mode)
- MOX, 5 batch, 50 MWd/kg burnup (forward mode)
- MOX, 5 batch, 50 MWd/kg target burnup (blending mode)
- Fast reactor, 6 batch, 180 MWd/kg burnup (forward mode)

The files can be found in the Bright-lite/examples/ folder. Run the following command to run the 42 MWd/kg burnup example case.

```
cyclus Bright-lite/examples/LWR42forward.xml
```

Library Interpolation

The libraries used in Bright-lite are often associated with several parameters. For example an LWR reactor library might have parameters for burnup, and enrichment. If as a user, you require a different value for these parameters there are two possible methods for obtaining it First, a new library can be generated externally from Bright-lite using tools available (XSGEN for example). It is also possible to create a dynamic library that matches your desired parameters using Bright-lite's built in library interpolation tool.

This tool is used using two key components in the Bright-lite input schema.

libraries - To enable library interpolation here simple add more than one library to the field. This is done simply by adding another val to the input field. That is...:

```
<val>extLWR</val>
```

represents a reactor library using just the *extLWR* library. However by adding another library:

```
<val>extLWR</val>

<val>lowLWR</val>
```

Bright-lite will make a new library based on the interpolation pairs and the values inside of these two libraries.

interpolation_pairs Once two or more libraries have been selected at least one interpolation pair will need to be added. An interpolation pair is a <"Parameter", Value> pair. The parameter represents a common parameter shared by the libraries, and the value is the target value for the new dynamic library in that parameter.

For example, there may be two LWR libraries that fit into an LWR library suite.

- Reactor 1

- Burnup: 50 MWd/kgIHM
- Enrichment: 5% U235

- Reactor 2

- Burnup: 30 MWd/kgIHM
- Enrichment: 3.3% U235

If a new library with the following parameters is desired

- Dynamic Reactor

- Burnup: 40 MWd/kgIHM
- Enrichment: 4% U235

The following xml should be added to the reactor archetype.

```
<libraries>
 <val>Reactor 1</val>
 <val>Reactor 2</val>
</libraries>
<interpolation_pairs>
 <key>BURNUP</key>
 <val>40</val>
 <key>ENRICHMENT</key>
 <val>4</val>
</interpolation_pairs>
```

Available Libraries

Recommended Libraries

- lowLWR - A standard PWR library.

- Enrichment: 2.2 %U235
- Burnup: 20 MWd/kgIHM
- PNL: 0.903

- Batches: 3

- standLWR

- Enrichment: 3.3 %U235
- Burnup: 33 MWd/kgIHM
- PNL:0.911
- Batches: 3

- extLWR

- Enrichment: 5% U235
- Burnup: 50 MWd/kgIHM
- PNL: 0.883
- Batches: 3

- FR25

- Burnup: 180 MWd/kgIHM
- Conversion Ratio: 0.25
- Batches: 6

- FR25MOX

- Burnup: 180 MWd/kgIHM
- Conversion Ratio: 0.25
- Batches: 6

- FR50

- Burnup: 180 MWd/kgIHM
- Conversion Ratio: 0.5
- Batches: 6

Additional Libraries

- E5_50

- Enrichment: 5% U235
- Burnup: 50 MWd/kgHM
- PNL: 0.965
- Batches: 3

- E5_60

- Enrichment: 5% U235
- Burnup: 60 MWd/kgIHM
- PNL: 0.98
- Batches: 3

- E7_100

- Enrichment: 7% U235
- Burnup: 100 MWd/kgIHM
- PNL: 0.974
- Batches: 5

- E9_100

- Enrichment: 9% U235
- Burnup: 100 MWd/kgIHM
- PNL: 0.981
- Batches: 4

Format of Reprocessing Plant Text File

```
BEGIN

isotope1n fraction1n

isotope2n fraction2n

...

isotopeN fractionN
```

```
END

BEGIN

isotope1k fraction1k

isotope2k fraction2k

...

isotopeK fractionK

END
```

References

[1] Bright Software, 2014. [Online]. Available: https://github.com/bright-dev/bright. [Accessed 1 2015].

[2] CNERG Fuel Cycle Group, "Cyclus," 01 01 2014. [Online]. Available: http://fuelcycle.org/. [Accessed 01 01 2014].

[3] M. Bell, "ORIGEN: the ORNL isotope generation and depletion code," Oak Ridge National Lab, Tenn. (USA), 1973.

[4] E. S. C. B. Robert Flanagan, "Multidimensional Cross Section Library Interpolation for the Bright-lite Reactor Modeling Software," *Global,* 2015.

[5] B. Lamarsh, Introductin to Nuclear Engineering, New Jersey: Prentice-Hall, 2001.

[6] M. Gidden, "An Agent-Based Modeling Framework and Application for the Generic Nuclear Fuel Cycle," 2013.

[7] J. J. Jacobson, G. E. Matthern, S. J. Piet and D. E. Shropshire, "VISION: Verifiable Fuel Cycle Simulation Model," in *ANFM*, 2009.

[8] E. A. Schneider, C. G. Bathke and M. R. James, "NFCSIM: A Dynamic Fuel Burnup and Fuel Cycle Simulation Tool," *Nuclear Technology,* vol. 151, 2005.

[9] L. Guerin, "Impact of Alternative Nuclear Fuel Cycle Options on Infrastructure and Fuel Requirements, Actinide and Waste Inventories, and Economics.," MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2009.

[10] R. Sanchez, I. Zmijarevic, M. Coste, E. Masielo, S. Santandrea, E. Martinolli, L. Villate, N. Schwartz and N. Guler, "APOLLO2 YEAR 2010," 2010.

[11] J. Grouiller, J. Vidal, A. Launay, Y. Berthion, A. Marc and H. Toubon, "CESAR: A Code for Nuclear Fuel and Waste Characterisation," in *WM*, Tucson, AZ, 2006.

[12] S. Yee, "Reactivity Worth Approach".

[13] OECD, "Physics and Safety of Transmutation Systems," Nuclear Energy Agency, 2006.

[14] R. Flanagan, C. Bagdatlioglu and E. Schneider, "Fuel Composition Techniques of Nuclear Fuel Cycle Simulators," in *Physor 2014 - The Role of Reactor Physics Toward a Sustainable Future*, Kyoto, Japan, 2014.

[15] Cyclus, "Cyclus Fuel Cycle Simulation," 2013. [Online]. Available: www.fuelcycle.org. [Accessed Feburary 2014].

[16] O. R. N. Laboratory, *Origen 2, Version 2.2,* Oak Ridge, Tennessee, June 2002.

[17] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *ACM Nat. Conf.*, 1968.

[18] W. Gorgon and J. Wixom, "Shepard's Method of "Metric Interpolation" to Bivariate and Multivariate Interpolation," *Mathematics of Computation Volume 32, Number 141,* pp. 253-264, 1978.

[19] MIT, "OpenMC User's Guide," MIT, 2014. [Online]. Available: http://mit-crpg.github.io/openmc/usersguide/index.html. [Accessed August 2014].

[20] K. S. Smith, "Assembly Homogenization Techniques for Light Water Reactor Analysis," *Progress in Nuclear Energy,* vol. 17, pp. 303-335, 1986.